

VŠB – Technická univerzita Ostrava  
Fakulta elektrotechniky a informatiky  
Katedra informatiky

**Informační systém pro správu  
konferencí a sborníků**

**Conference and Workshop  
Management System**

## Zadání diplomové práce

Student:

**Bc. Lukáš Zátpek**

Studijní program:

N2647 Informační a komunikační technologie

Studijní obor:

2612T025 Informatika a výpočetní technika

Téma:

Informační systém pro správu konferencí a sborníků  
Conference and Workshop Management System

Jazyk vypracování:

čeština

Zásady pro vypracování:

Existující systémy pro správu konferencí většinou obsahují pouze podporu správy konferencí a recenzního řízení článků, nikoliv podporu tvorby sborníku. V akademickém roce 2015/2016 byl v rámci bakalářské práce vytvořen webový informační systém nazvaný Confee, který umožňuje jak správu konferencí a recenzního řízení článků tak tvorbu sborníků konferencí. Informační systém Confee vykazuje některé nedostatky, např. sborníky mohou být vytvořeny pouze pro styl LNCS, systém obsahuje pouze triviální subsystém pro tvorbu recenzí, ořezy článků jsou pevně dané, atd. Cílem práce je dopracovat zjištěné nedostatky a rozšířit systém pro všechny dostupné formáty sborníků.

Úkoly:

1. Doladění zjištěných nedostatků a jejich řádné otestování.
2. Vytvoření subsystémů pro tvorbu recenzí a správu sborníků.
4. Rozšíření systému o tvorbu šablon pro ořezy článků.
5. Optimalizace datové vrstvy.
6. Návrh a implementace nové prezentační vrstvy.
7. Vytvoření online nápovědy.
8. Zhodnocení systému po nasazení do ostrého provozu, porovnání s existujícími systémy.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího diplomové práce.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Peter Chovanec, Ph.D.**

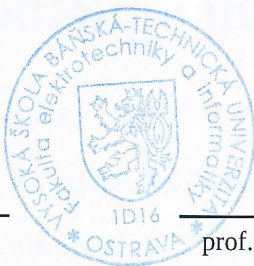
Datum zadání: 01.09.2017

Datum odevzdání: 30.04.2018



---

doc. Ing. Jan Platoš, Ph.D.  
vedoucí katedry



---

prof. Ing. Pavel Brandštetter, CSc.  
děkan fakulty



Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 30. dubna 2018

*Lukáš Holý*  
.....



Rád bych poděkoval Ing. Petrovi Chovancovi, Ph.D., za odborné vedení mé diplomové práce, velkou ochotu a přínosné konzultace, které proběhly v rámci realizace této práce. Také chci poděkovat své rodině a přátelům, za trpělivost, vstřícnost a nezměrnou podporu.

## Abstrakt

Diplomová práce má za cíl navrhnout a naimplementovat informační systém, jehož účelem je správa konferencí a sborníku s podporou recenzního řízení. Samotným organizátorům je umožněno vytvořit a spravovat sborník v rámci open-source (ShareLaTeX) řešení, které je vhodným způsobem zaintegrováno do systému, a jenž poskytuje úplnou funkcionalitu pro správu sborníku. Diplomová práce se v prvotní fázi soustředí na popis kontinuity na bakalářskou práci, na kterou nepřímo navazuje. Provádí porovnání s aktuálními systémy a popisuje rozhodnutí, které vedly k výběru daných technologií. Druhá fáze práce se zabývá podrobnou analýzou celého informačního systému včetně navržené architektury s definicí použitých primárních technologií a nástrojů. Poslední fáze práce je věnována samotné implementaci, popisu důležitých algoritmů v systému, definování uživatelského rozhraní a věnování se možným vizím a rozšířením do budoucna.

**Klíčová slova:** L<sup>A</sup>T<sub>E</sub>X, ShareLaTeX, Informační systém, Konference, Sborník, ASP.NET Core, Angular 5

## Abstract

The aim of this thesis is to design and implement an information system with the purpose of managing conferences and proceedings via a review process. The organizers are able to create and manage almanacs through an open-source (ShareLaTeX) solution that is appropriately integrated in the system and provides full functionality for the management of proceedings. In the first part, the thesis elaborates on the bachelor thesis, indirectly mirroring its topic. The thesis provides a comparison with current systems in use, and describes decisions that have led to the selection of particular technologies. The second part of the thesis deals with a detailed analysis of the entire information system including the proposed architecture with the definition of predominantly applied technologies and tools. The final part of the thesis is focused on the actual implementation, description of important algorithms in the system, definition of the user interface, and pays attention to possible future improvements and extensions.

**Key Words:** L<sup>A</sup>T<sub>E</sub>X, ShareLaTeX, Information system, Conference, Proceedings, ASP.NET Core, Angular 5

# Obsah

Seznam použitých zkratk a symbolů	9
Seznam obrázků	10
Seznam tabulek	11
Seznam výpisů zdrojového kódu	12
<b>1 Úvod</b>	<b>13</b>
<b>2 Existující systémy</b>	<b>15</b>
2.1 OpenConf . . . . .	15
2.2 EasyChair . . . . .	19
2.3 Zhodnocení . . . . .	22
<b>3 Confee 1.0</b>	<b>23</b>
3.1 Seznámení se systémem . . . . .	23
3.2 Funkcionalita . . . . .	28
3.3 Nedostatky . . . . .	29
<b>4 Confee 2.0 - Analýza systému</b>	<b>30</b>
4.1 Datová analýza . . . . .	30
4.1.1 Informační systém pro správu konferencí a sborníků . . . . .	30
4.1.2 ShareLaTeX . . . . .	40
4.2 Funkční analýza . . . . .	41
4.2.1 Architektura systému . . . . .	41
4.2.2 Stavy a role v systému . . . . .	43
4.2.3 Funkcionalita systému . . . . .	44
4.2.4 Subsystem pro tvorbu ořezů a stylů . . . . .	50
4.2.5 Subsystem pro tvorbu posudků . . . . .	51
<b>5 Confee 2.0 - Použité technologie a nástroje</b>	<b>52</b>
5.1 Úvod . . . . .	52
5.2 Prezentační vrstva . . . . .	53
5.3 Aplikační vrstva . . . . .	55
5.4 ShareLaTeX a Docker . . . . .	57
5.5 Technologie . . . . .	58
5.6 Nástroje . . . . .	59



<b>6</b>	<b>Confee 2.0 - Implementace</b>	<b>61</b>
6.1	Registrace a správa uživatelů . . . . .	61
6.2	Správa konferencí . . . . .	62
6.3	Zpracování PDF dokumentu . . . . .	63
6.4	Tvorba sborníku . . . . .	66
6.5	Tvorba recenzí . . . . .	68
<b>7</b>	<b>Confee 2.0 - Uživatelské rozhraní</b>	<b>70</b>
<b>8</b>	<b>Závěr</b>	<b>74</b>
	<b>Literatura</b>	<b>76</b>
	<b>Přílohy</b>	<b>79</b>
<b>A</b>	<b>CD obsahující</b>	<b>80</b>

## Seznam použitých zkratek a symbolů

UI	– User Interface
SPA	– Single Page Application
UX	– User Experience
JS	– JavaScript
AJAX	– Asynchronous JavaScript and XML
PDF	– Portable Document Format
BAT	– Batch File
IS	– Informační systém
JSON	– JavaScript Object Notation
BSON	– Binary JSON
IDE	– Vývojové prostředí
API	– Application Programming Interface
ORM	– Object Relational Mapper
HTML	– HyperText Markup Language
CSS	– Cascading Style Sheets
LESS	– Leaner Style Sheets
OS	– Operační systém
SaaS	– Software as a Service
CLI	– Command Line Interface
DI	– Dependency Injection
YAML	– Ain't Markup Language
JWT	– JSON Web Token
HMAC	– Keyed-hash Message Authentication Code
LNCS	– Lecture Notes in Computer Science
WYSIWYG	– What You See Is What You Get
IDE	– Integrated Development Environment

## Seznam obrázků

1	Nastavení dostupných modulů v IS OpenConf . . . . .	16
2	Homepage stránka IS OpenConf . . . . .	17
3	Část upload fomuláře pro nahrání příspěvku v IS OpenConf . . . . .	18
4	Nastavení sborníku v IS OpenConf . . . . .	19
5	Žádost o instalaci EasyChair . . . . .	20
6	Confee 1.0 nahrání příspěvku . . . . .	24
7	Confee 1.0 založení konference . . . . .	25
8	Confee 1.0 vytvoření posudku . . . . .	26
9	Confee 1.0 subsystém pro ořezy příspěvků . . . . .	27
10	Confee 1.0 editace LaTeX souboru . . . . .	28
11	ERD - logický model databáze . . . . .	31
12	MongoDB jednoduché schéma pro popis důležitých kolekcí . . . . .	41
13	Schéma popisující architekturu systému . . . . .	42
14	Aktivitní diagram popisující zařazení publikace do sborníku . . . . .	49
15	Logo Confee 2.0 . . . . .	70
16	Administrátorské rozhraní systému Confee 2.0 . . . . .	70
17	Administrátorské rozhraní systému Confee 2.0 . . . . .	71
18	Uživatelské rozhraní systému Confee 2.0 . . . . .	72
19	Online dokumentace systému Confee 2.0 . . . . .	73
20	Online dokumentace systému Confee 2.0 . . . . .	73



## Seznam tabulek

1	Schéma tabulky Style . . . . .	33
2	Schéma tabulky ReviewResult . . . . .	33
3	Schéma tabulky AspNetUsers . . . . .	34
4	Schéma tabulky Conference . . . . .	35
5	Schéma tabulky StyleCropping . . . . .	36
6	Schéma tabulky StyleDocument . . . . .	36
7	Schéma tabulky Page . . . . .	37
8	Schéma tabulky MenuItem . . . . .	37
9	Schéma tabulky Almanac . . . . .	37
10	Schéma tabulky Review . . . . .	38
11	Schéma tabulky Publication . . . . .	38
12	Schéma tabulky PublicationFileResult . . . . .	38
13	Schéma tabulky Author . . . . .	39
14	Schéma tabulky Comment . . . . .	39
15	Schéma tabulky AspNetRoles . . . . .	39

## Seznam výpisů zdrojového kódu

1	Vytvoření projektu pomocí Angular CLI . . . . .	54
2	Vytvoření komponenty pomocí Angular CLI . . . . .	55
3	Konfigurační soubor ve formátu JSON . . . . .	56
4	ORM Eager-loading . . . . .	56
5	YAML soubor pro konfiguraci ShareLaTeX služby . . . . .	57
6	Provedení instalace služby v rámci Docker technologie . . . . .	57
7	Vkládání JWT tokenů do hlaviček HTTP požadavků . . . . .	61
8	Vytvoření nové konference . . . . .	63
9	Provedení ořezu PDF dokumentu . . . . .	65
10	Vložení záznamu do databáze MongoDB . . . . .	67
11	Přiřazování oponentů k publikaci . . . . .	69

# 1 Úvod

Informačních systémů pro správu konferencí s podporou recenzních řízení je k nalezení v online prostředí velké množství. Mnohdy však tyto systémy postrádají možnost spravovat vytvářené sborníky tvořené z nahraných příspěvků k vybrané konferenci. Často sdílí značné spektrum podobných funkcionalit, rozdíl pak mnohdy bývá v úrovni samotné implementace, v kvalitě provedení user-interface a user-experience. Tyto aspekty však velice často rozhodují o úspěchu či neúspěchu dané společnosti – bavíme-li se v rovině poskytovaných služeb v korporátním prostředí formou software as a service, dále jen SaaS.

Vysoká škola báňská – Technická univerzita Ostrava je častým pořadatelem konferencí a k těmto účelům využívá vlastní informační systém pro správu konferencí s podporou správy sborníku. Abychom pochopili celý proces správy konferencí a dokázali si tak vytvořit nějaký mentální model toho, co vše tento proces obnáší, popíšeme si následující kroky, které vedou až k vytvoření samotného sborníku.

K vybrané konferenci je uživatelům umožněno přispívat svým odborným článkem – ten se skládá z PDF dokumentu a informací o daném článku (kontaktní autor, autoři, abstract a podobně). Takovýto článek je pak v rámci recenzního řízení zkontrolován, tedy zdali obsah textu a kvalita zpracování odpovídá požadovaný standardům. Na základě tohoto recenzního řízení vzniká posudek určující splnění respektive nesplnění stanovených podmínek. V následujícím kroku správce konference provádí činnost pro kontrolu formátu příspěvku. Jednotlivé nahrané příspěvky (respektive PDF dokumenty) musí odpovídat definovanému stylu, který je při vytváření konference jednoznačně určen (v aktuálním systému pouze LNCS). Pakliže daný příspěvek byl v recenzním řízení schválen, prošel kontrolou formátu, může být zařazen do sborníku. Vytvoření sborníku tedy spočívá ve vytvoření PDF dokumentu obsahující všechny přijaté příspěvky. Aktuální systém má však řadu nedostatků. Řada funkcionalit musí být prováděna samotným správcem bez větší míry automatizace, celková správa konference je místy značně nepřehledná a zdlouhavá. Jsou použité technologie, které jsou již zastaralé a pro moderní potřeby nedostačující. Zavedení nových funkcionalit do systému je obtížné. Ke správě sborníku chybí komplexní podsystém, vše je řešeno pouze pomocí příkazové řádky s minimální podporou pro správu chyb.

V akademickém roce 2015/2016 vznikla myšlenka, vytvoření nového informačního systému, který si dával za cíl tyto jednotlivé problémy značným způsobem vyřešit. V podobě bakalářské práce vznikl tak nový informační systém pro správu konferencí a sborníků Confee 1.0. Tento systém se zejména soustředil na podporu tvorbu sborníku, zavedení automatizace do systému a zpřehlednění jednotlivých činností obsažené při správě konferencí. Při uvedení tohoto systému do provozu jsme však narazili na řadu nedostatků, jejichž přítomnost samotný proces správy konference a sborníku znesnadnily.

Prvotním cílem této diplomové práce je navázání na systém předešlý, provedení zjištění nedostatků a jejich následnou opravu, aktuální funkcionalitu vylepšit nebo zcela doplnit o novou. Avšak v rámci prvotních analýz jsme dospěli k následujícímu závěru. Stav, ve kterém se systém



Confée 1.0 náchází, nedovoluje vhodným způsobem systém rozvíjet, aplikovat nové postupy a technologie. Přičemž objevení právě nových technologií nás přesvědčilo o možnosti tento systém zpracovat na lepší úrovni. Toto rozhodnutí nás tedy vede k vystavění zcela nového informačního systému postaveného na nové architektuře za použití moderních technologií umožňující snadnější udržitelnost a rozšiřitelnost. Na nově vyvíjený systém se pokusíme nahlížet jako na celek skládající se z jednotlivých podsystémů, přičemž právě tyto podsystémy jsou důležitými částmi při správě konferencí a sborníků. Naše pozornost se bude soustředit na výrazné zlepšení podpory pro správu sborníků v kooperaci s možnostmi, které jsou nabízeny řadou open-source řešení. Cílem se tak stala tvorba informačního systému postaveného na zcela nové architektuře podporující celý proces správy konferencí a sborníku v integraci s vyvinutými subsystemy a integraci open-source řešení, se zásadně změněným uživatelským rozhraním podporující snadnou orientaci a ovládatelnost systému a za podpory vytvoření online dokumentace sloužící jako manuál pro nové uživatele. Nově vyvinutý informační systém pro správu konferencí a sborníku tak ponese označení Confée 2.0.

V první kapitole v rámci krátké rešerše představíme existující systémy a detailně rozebereme některé z nich. V následující kapitole se seznámíme s datovým pohledem nad systémem Confée 2.0. Popíšeme vazby mezi jednotlivými entitami a jejich atributy. Seznámíme se s funkčními požadavky a detailně je popíšeme. Zavedeme jednotlivé role a jejich působnost v systému Confée 2.0. Rozebereme jednotlivé subsystemy a probereme celkovou navrženou architekturu systému společně s popisem jednotlivých funkcionalit v systému. Další kapitola se věnuje popisu použitých technologií, kdy se zaměříme zejména na ty nosné. V kapitole Implementace analyzujeme některé z funkcionalit, představeny budou v příloženém výpisu kódu společně s popisem. Rovněž krátkou kapitolu věnujeme grafickému rozhraní, které prošlo razantní úpravou. V kapitole Závěr poté zhodnotíme naše výsledky. Rozebereme případné nástrahy, které nás při vývoji systému potkaly. Zamyslíme se nad možnými vylepšeními. Načrtneme vizi, jak by se systém mohl v budoucnu ubírat.

## 2 Existující systémy

V prostředí informačních systémů zabývajících se správou konferencí je k naleznutí mnoho systému pro tuto potřebu, vyjmenujme si například Online Registry [8], Converia [9], OpenConf [10], COMS [11], ConfTool [12] a EasyChair [13], přičemž právě zmiňovaný EasyChair bychom mohli považovat za jeden z nejznámějších systémů používaného pro správu konferencí v IT prostředí. Funkcionalita těchto systémů je v řadě případů velice podobná. V rámci spravování konferencí je určitě zcela nezbytné, aby organizátoři dokázali konferenci vytvořit a spravovat, to mimo jiné znamená – správa jednotlivých stránek, registrace a přihlašování uživatelů, správa nahrávaných dat a další. Mnoho těchto funkcionalit se v těchto systémech prolíná a pak již tedy jen zaleží na implementační úrovni reprezentující, jak moc je daná funkcionalita zdařilá. Následující odstavce bychom rádi věnovali detailnějšímu popisu vybraných informačních systémů pro správu konferencí a recenzního řízení.

### 2.1 OpenConf

OpenConf [10] je informační systém, jehož původním záměrem byla podpora a správa konferencí. Avšak postupem času došlo k přizpůsobení i pro jiné typy aktivit – časopisy, workshopy, sympozia. Systém je provozován ve vícero než 100 zemích po celém světě [38]. Je dodáván ve třech možných edicích, přičemž jsou od sebe odlišeny dostupnými funkcemi v závislosti na možnosti za danou službu platit. OpenConf však jednu ze svých verzí nabízí zdarma – Community Edition. Nicméně je nutné splnit určité podmínky, a to použití tohoto softwaru pouze na vlastních webových stránkách a pouze pokud v souvislosti s používaným softwarem nebo službami nebude docházet k žádné platbě. V ostatních případech je již nutné zaplatit licenční poplatek.

Systém je nutné nainstalovat lokálně na vlastní server. Pouze v případě nejbohatší edice je systém založen na principu využití služby SaaS a zákazníkovi je umožněn přístup ke službě v rámci internetu. V ostatních případech je provedení vlastní instalace nutné, což může být pro neznalé uživatele náročné. OpenConf je napsán v PHP [41] využívající databázový server MySQL [42] nebo MariaDB [43], je tedy vhodné, aby člověk provádějící instalaci měl základní znalosti těchto technologií. Při spuštění instalace pak dochází k nastavení řady parametrů, které ovšem systém následně umožňuje editovat po přihlášení do systému. Uvedme si pár atributů dostupných přímo z online dokumentace na webové stránce OpenConf [34]:

`OCC_SESSION_VAR_NAME` – The session variable name to use for this installation of OpenConf. If you have multiple installations of OpenConf running on the server, you should make this unique for each installation.

`OCC_INSTALL_COMPLETE` – Tracks whether the installation of OpenConf completed. After installing OpenConf, to re-install via the web interface, you will need to change this to 0.

OCC\_CHAIR\_PWD\_TRUMPS – Allows the Chair to sign in as an author or reviewer using the Chair's password.

OCC\_ADVANCED\_CONFIG – Enables access to the advanced configuration settings by the Chair.

OCC\_DB\_USER, OCC\_DB\_PASSWORD, OCC\_DB\_HOST, OCC\_DB\_NAME, OCC\_DB\_PORT – Database settings.

OCC\_DB\_PREFIX – Database table prefix, useful when multiple installations of OpenConf are in the same database.

OpenConf poskytuje tzv. moduly, ty slouží k přidávání funkcionality nad rámec hlavních možností systému (samotná instalace systému již poskytuje nějakou základní funkcionality). Jednotlivé moduly lze pak instalovat pomocí administrátorského rozhraní, přičemž v reálném čase se jednotlivé moduly dají aktivovat případně deaktivovat. Množství a volba modulů závisí na volbě edice. V případě zmíněné Community Edition jsou dostupné pouze tři moduly – CAPTCHA, kontrola nahrávaného formátu, Mailová služba. Pak již záleží na samotném uživateli, zdali se rozhodne pro rozšíření v podobě zakoupení lepší edice. Forma rozšiřování modulu je jistě zajímavou volbou, jak uspokojit specifické potřeby náročných klientů, aniž by se muselo razantně zasahovat do systému.

OpenConf Conference 2020

OpenConf Intranet & Conference Management System v6.x

Chair Home Help Sign Out Signed in as: chair

Modules

Additional OpenConf functionality is available through modules. In order to use a module, you must first install it, and then activate it. If a module is installed, but not active, its data will still be preserved. Uninstalling a module will result in the module's data being deleted. The Module Home link provides access to the module's configuration settings. The Version link provides information about the module.

Status Legend: ■ Active ■ Disabled ■ Not Installed

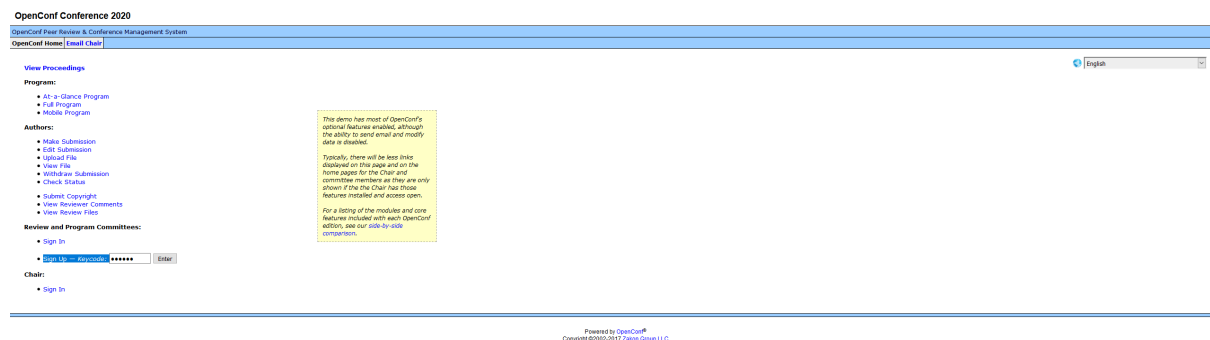
Module Name	Version	Source	Description	Action
Acceptance	1.1.0	OpenConf	Provides Chair with the ability to change acceptance options	config deactivate uninstall
Advocate Assign	2.0.2	OpenConf	Allows advocates to assign reviews	config deactivate uninstall
Auto Assign	1.0.0	OpenConf	Automatically assigns reviewers/advocate when a submission is made	config deactivate uninstall
Bidding	1.7.0	OpenConf	Bidding on papers by reviewers, and bid-based assignments	config deactivate uninstall
CAPTCHA	2.0.0	OpenConf	Checks that submissions are being made by a human	config activate uninstall
Copyright Act	2.0.1	OpenConf	Provides report of submissions in ACTE/CPD/CSW format	config deactivate uninstall
Copyright-BSE	2.1.1	OpenConf	Provides author referral form to BSE's electronic copyright system	config deactivate uninstall
Custom Forms	1.5.1	OpenConf	Customize submission, review, and committee profile forms	config deactivate uninstall
Demo	1.0.0	OpenConf		deactivate uninstall
Discussion	1.0.2	OpenConf	Online discussion (forum) for committee members	config deactivate uninstall
File Format Check	3.0.1	OpenConf	Verifies file uploaded matches selected format	config deactivate uninstall
Form Fields	1.0.0	OpenConf	Select fields to include on submission and review forms	config deactivate uninstall
Mail	1.0.0	OpenConf	Provides support for using an external SMTP server	config deactivate uninstall
Multiple	1.5.1	OpenConf	Multiple file type uploads	config deactivate uninstall
Payment	4.0.0	OpenConf	Payment libraries used by other modules	config deactivate uninstall
Plagiarism-Detect	1.0.1	OpenConf	Docotic plagiarism checking service	config deactivate uninstall
Plagiarism-Authenticate	1.1.1	OpenConf	CrossCheck and iThenticate plagiarism checking services	config deactivate uninstall
Proceedings	2.1.0	OpenConf	Online proceedings	config deactivate uninstall
Program	4.2.0	OpenConf	Online and mobile program building and display options	config deactivate uninstall
Publons	1.0.0	OpenConf	Provides review credit to committee members via Publons service	config deactivate uninstall
Rebuttal	1.7.0	OpenConf	Author rebuttal of reviews, and reviewer rebuttal of author comments	config deactivate uninstall
Reviewer Upload	2.2.0	OpenConf	Reviewer file upload for assigned reviews (e.g., annotation, feedback)	config deactivate uninstall
Sub-Pre-Payment	2.2.0	OpenConf	Require payment for making a submission	config activate uninstall

Powered by OpenConf®  
Copyright ©2002-2017 Zalon Group LLC

Obrázek 1: Nastavení dostupných modulů v IS OpenConf

Velký nedostatek vnímáme v oblasti podoby grafického rozhraní. Jedná se o klasický případ vývoje softwaru, který rostl s dobou. Tento systém je na trhu od roku 2002. Není pochyb, že s rostoucími klienty rostou i požadavky na nové funkcionality. Rostou však požadavky i na vývoj modernějšího UI? Nabyli jsme dojmu, že OpenConf je vlastně informační systém, který má sloužit aktuálním klientům a ty nové spíše odradit, než přilákat. Pakliže nejste nějak seznámení s mentálním modelem fungování samotné správy konferencí, budete mít velký problém

se mezi jednotlivými položkami UI zorientovat. Skončíte pak dlouhým studováním grafických částí a budete se snažit přimět systém, aby naslouchal vašim požadavkům. Dobré UI je totiž o schopnosti pomoci lidem orientovat se v nových systémech. Pomáhá jim tyto systémy poznávat, učit se a následně je kvalitně využívat. V moderních systémech pak často kvalita UI staví korporace na propast mezi úspěchem a neúspěchem, přičemž myšlenka samotného projektu může být účtyhodná.



Obrázek 2: Homepage stránka IS OpenConf

Pro otestování jednotlivých funkcionalit jsme využili dostupné demo [39], které obsahuje téměř veškeré funkce poskytované nejvyšší edicí. Nebudeme popisovat veškeré funkcionality, spíše se zaměříme na ty, které nás zaujaly, na správu konference a samotné nahrávání publikace.

V případě, že se uživatel rozhodne nahrát svůj příspěvek, je nutné, aby vyplnil příslušný formulář. Ten se skládá z následujících skupin - Autoři, kontaktní autor, zaměření příspěvku, obsah společně s nahrávaným souborem, nastavení hesla a přidání doplňujícího komentáře. Přičemž kontaktnímu uživateli je pak již jenom zasláno ID (identifikátor jeho příspěvku) na uvedenou emailovou adresu. Pakliže chce provést nějaké změny, přihlásí se pod tímto jednoznačným identifikátorem společně s vyplněním jeho hesla. Za zmínku stojí právě neexistence uživatelského profilu, v rámci kterého by se uživatel mohl přihlásit. Všechny funkcionality pro správu svého profilu by měl následně seskupené pod jedním „regionem“. Uživatel tak vždy musí vybrat, jakou změnu chce provést (změna hesla, nastavení profilu, změna příspěvku) a vždy je donucen zadat své přihlašovací údaje a po úspěšném přihlášení je mu funkcionalita zpřístupněna. Ostatně tohoto faktu jsme si mohli povšimnout i u ostatních informačních systémů zabývajících se správou konferencí a příspěvků. Model těchto systémů je nastaven tak, že většinou neexistuje uživatelský profil, jak jsme navykli třeba při nákupu na e-commerce portálech. Pravdou je, že my jsme tento model narušili a vydali jsme se cestou seskupení všech uživatelských funkcionalit pod profil daného uživatele. Jaký dopad to bude mít na chování uživatelů a orientaci na webovém rozhraní je již pak otázka testovacích procesů.

Po přihlášení do systému v rámci role jako předsedající (můžeme považovat za roli administrátorského typu) je k dispozici široká nabídka dostupných funkcionalit. Dají se rozdělit do

OpenConf Conference 2020  
OpenConf Peer Review & Conference Management System  
[OpenConf Home](#) [Email Chair](#)

Submission

General Information

Submission Title:

Submission Type: ☒ Paper ☐ Poster

Student: ☐ Yes ☒ No

Author(s)

Author 1

ORCID:

Isosch:

First/Given Name:

Last/Family Name:

Suffix:

Position/Title:

Organization:

Department/Division:

Address:

Address 2:

City:

State/Province:

Postcode/Zip:

Country:

Email:

Telephone:

Web Site:

Facebook Page:

Google+ Page:

Twitter Page:

LinkedIn Page:

Photo Link:

Presenter: ☐ Yes ☒ No

Biography:

This demo form is displaying all built-in fields available in OpenConf. The Plus Edition's Form Fields module allows these fields to be overridden. The Professional Edition's Custom Forms module allows these fields to be overridden, new fields added, and all fields re-ordered.

[view community edition form \(linked page to view\)](#)

[Add Author](#)

Contact Author

Contact Author:  Author 1 ☒

Author who will serve as the point of contact for correspondence about the submission.

Alternate Contact:  [ipamela@centrum.cz](#)

Obrázek 3: Část upload fomuláře pro nahrání příspěvku v IS OpenConf

následujících skupin.

- **Obecné nastavení**

V této kategorie se dá naleznout obecné nastavení v rámci vytvářené konference. Co nás zaujalo je poměrně zajímavý konfigurační management umožňující nastavení různých parametrů dle uživatelských preferencí. OpenConf vám tak umožní v rámci jednoduché administrace nastavit třeba šablony pro odesílání emailů, nastavení flagů (true nebo false) pro jednotlivé funkce a další. V této kategorii se nachází i již zmiňované nastavení modulů. Nesmíme opomenout i poměrné velké množství lokalizačních možností, ne tedy jenom angličtina.

- **Příspěvky**

Obsahuje běžné funkcionality pro práci s příspěvků. K naleznutí je detailní popis jednotlivých příspěvků, které byly nahrány. Lze je samozřejmě editovat, anebo případně vyexportovat jmenný seznam do formátu csv. Správci je umožněno nastavit, které atributy se mají do csv vložit. Co jsme shledali jako zajímavé, je kontrola plagiátorství, a to zejména v případě velkých konferencí, kde množství příspěvků může být opravdu velké. V OpenConf tuto potřebu řeší pomocí napojení na externí systém iThenticate [40].

- **Porota**

Správa recenzentů, jenž jsou zodpovědní za vytváření posudků.

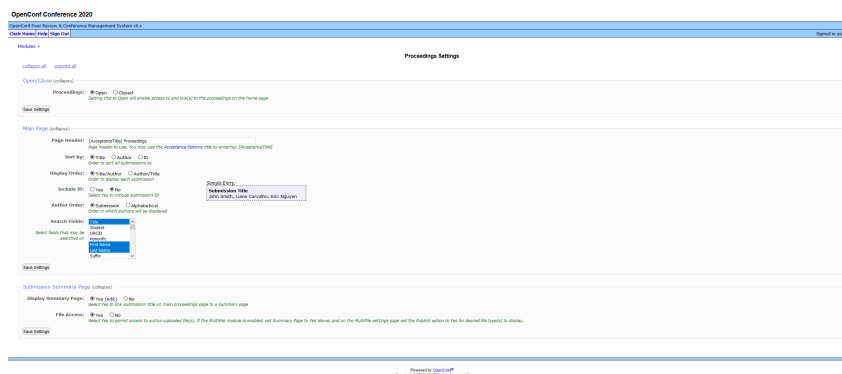
- **Posudky a celkové skóre**

Kategorie pro správu posudků a rovněž pro správu celkového skóre, které tyto posudky

nad přijatými příspěvky generují. Zajímavou možností nám přišlo nastavení automatického přiřazení posudku k oponentovi. Systém pak na základě algoritmu vybere vhodného oponenta. Co rovněž oceňujeme je možnost nastavení primárního zaměření pro každého oponenta, přičemž systém pak tuto skutečnost bere v potaz a snaží se párovat pouze ty příspěvky, které odpovídají nastaveným preferencím.

- **Sborník**

Přijaté příspěvky jsou přidávány do sborníku. V Confee 2.0 pojetí, se sborník skládá ze sady publikací a LaTeXových dokumentů, ty jsou kompilovány a výsledkem je jeden PDF dokument. Ten odpovídá předem definovanému stylu. V případě informačního systému OpenConf, dochází k přidávání přijatých příspěvků do seznamu, který je pak dostupný prostřednictvím webového rozhraní. K dispozici je však i externí modul pro podporu IEEE příspěvků.



Obrázek 4: Nastavení sborníku v IS OpenConf

- **Stránky**

Funkcionalita podporující dynamické vytváření obsahu pro jednotlivé stránky. Ty jsou následně propojeny systémem odkazů. V Confee 2.0 využíváme WYSIWYG editoru TinyMce [44], přičemž OpenConf využívá CkEditor [45].

## 2.2 EasyChair

EasyChair [13] je informační systém určený pro správu konferencí, který je flexibilní, snadno použitelný a má poměrně mnoho funkcí, které můžou tak uspokojit potřeby různé typy konferenčních modelů [37]. EasyChair je nabízen ve čtyřech licenčních variantách, kdy dostupnost funkcionalita je stupňována na základě volby konkrétní licence a míry vynaložených finančních prostředků. Nabízené licence jsou následující – zdarma, profesionální, korporátní a skupinová.

EasyChair poskytuje své služby formou SaaS. To může být velká výhoda pro nové zákazníky, kteří nechtějí řešit žádné instalační procesy nutné ke spuštění svého systému. Tento přístup k

softwaru jako službě, platí pro všechny poskytované licence, zatímco systém OpenConf tuto variantu podporuje jen pro svou nejvýše placenou licenci. Jedinou možností jak otestovat funkcionality poskytovanou pro správu konferencí je vytvoření žádosti o provedení nové instalace konference. Možnost veřejného dema by byla řadou uživatelů jistě vřele přijata. Po odeslání požadavku je potřeba vyčkat na potvrzující zprávu ze strany EasyChair o schválení vytvářené konference.

Obrázek 5: Žádost o instalaci EasyChair

Následuje rozdělení funkcionality do několika skupin.

## 1. Hosting a podpora

Jak jsme již nastínili na začátku této kapitoly, služby spojené s EasyChair jsou poskytovány formou online služby. Klient nemusí nic instalovat a k využití služby mu tak stačí pouze internetové připojení. V případě poruchy a následného pádu systému, jsou vaše data zálohovány a vy tak máte jistotu, že o své data nepřijdete. EasyChair nabízí Helpdesk, který je vám k dispozici v případě problému. Nicméně tato funkcionality je dostupná pouze pro licence *korporátní a skupina*.

## 2. Různé

Zajímavá funkcionality, kterou systém poskytuje, je možnost importovat konfiguraci a program z jedné konference do druhé, a to může ušetřit vzácný čas a eliminovat výskyt případných chyb.

## 3. Autoři a příspěvky

Samotný proces nahrávání příspěvků poskytuje běžné funkcionality vyskytující se u konferenčních systémů. Nahrání vícero souborů, zasílání notifikací v případě přijetí nebo zamítnutí příspěvku, umožnění stáhnutí předem připravených šablon pro snadnější tvorbu příspěvku. Nicméně jedná funkcionality, se z daného standardu vymyká. Umožnění vytváření upravitelných formulářů – v případě, že potřebujete v rámci specifického konferenčního modelu přidat položky v procesu nahrávání příspěvku, které nejsou běžné.

## 4. Porota a posudky

V této skupině najdeme celou řadu funkcionalit – možnost online diskuze přijatých příspěvků s ostatními členy poroty. Jednotliví recenzenti mají možnost provést nastavení svých preferencí v rámci zaměření, kterému se věnují. Na základě těchto preferencí jsou jim přidělovány jednotlivé příspěvky. Automatické přidělování dostupných recenzentů je přítomno, stejně tak jako v systému OpenConf. Vzhledem k tomu, že kritéria pro přijetí daného příspěvku mohou být pro jednotlivé konferenční modely různé, nabízí EasyChair zajímavou funkcionality. Stejně jako v případě skupiny *Autoři a příspěvky* systém dovoluje vytvářet upravitelné formuláře, na základě kterých vznikají závěrečné posudky. Systém zároveň sbírá statistiky ohledně vytvořených posudků a jednotlivých recenzentech a je schopen je nabídnout ve formě textového nebo grafického shrnutí.

## 5. Data export

Veškerá data vytvořená v průběhu správy konference lze exportovat do formátu typu XLSX, CSV.

## 6. Email management

V rámci správy emailové komunikace byl vytvořen subsystém pro správu emailů. Ten dovoluje vytvářet emailové šablony. Jestliže uživatelé zasílají emaily prostřednictvím EasyChair, je systém schopen tyto emaily logovat a využít pro pozdější potřeby.

## 7. Sborník

EasyChair nabízí jeden styl sborníku ve formátu LNCS. V případě, že nechcete studovat pokyny pro tvorbu určené editorům v rámci organizace Springer, je EasyChair schopen vytvořit sborník jako jeden zip soubor, který můžete pak v rámci této organizace nahrát.

## 8. Program a správa stránek

Tento podsystém není jako jediný dostupný pro licenci *zdarma*. Umožňuje správu obsahu



jednotlivých stránek vytvářených při zakládání konference, společně s možností voleb publikace sborníku. Ve vyšší licenci je uživateli umožněno stahovat dostupné šablony, které jsou poskytované systémem EasyChair.

EasyChair podobně jako OpenConf trpí špatně zpracovaným UI. Noví uživatelé mají velké obtíže orientovat se v systému. Prostředí je neintuitivní a vyhledání potřebné funkcionality zabere mnohdy spoustu času.

## 2.3 Zhodnocení

Řada informačních systémů pro správu konferencí a recenzního řízení vznikala v době, kdy technologie nebyly na tak vysoké úrovni. Neřešila si příliš podoba uživatelského rozhraní. Jedinou možností, jak vystavět webovou stránku, bylo pomocí tabulek. Uživatelé, kteří nejsou ve správě konferencí zaběhnutí, mohou velice rychle propadnout skepsi při vyhledávání požadované funkcionality. Funkcionalitou jsou tyto systémy naplněné. Umožňují velké množství akcí a nastavení, které jsou možné v rámci konference provádět. Některé z těchto systémů poskytují specifitější funkce oproti druhým. Avšak obecně můžeme říci, že se pohybujeme pouze v oblasti jednotek rozdílů. Ve své podstatě jsou si funkcionálně tyto systémy velmi blízké. Nenarazili jsme na žádný informační systém pro správu konferencí, který by plně umožňoval spravovat sborník. Řada těchto systémů umožňuje spravování sborníku pouze v podobě jednoho stylu, a to ještě velmi omezeným stylem. Subsystem pro správu LaTeXových souborů jsme nezaznamenali v žádném pozorovaném informačním systému.

## 3 Confee 1.0

### 3.1 Seznámení se systémem

S původní myšlenkou pro správu konferencí a sborníku přichází informační systém vytvořený v akademickém roce 2015/2016 v rámci bakalářské práce. Tento vytvořený systém si kladl za cíl vyřešit nedostatky předchozího systému, který do celého procesu správy vnášel minimální míru automatizace a samotná správa v řadě momentech působila velice chaoticky. V následujících odstavcích obecně představme funkcionalitu, kterou Confee 1.0 svým koncovým uživatelům nabízí.

Systém bychom mohli rozdělit na dvě části, část neveřejná a veřejná. Veřejná část je dostupná všem koncovým uživatelům, u kterých není vyžadováno přihlášení do systému a je jim poskytnuta funkcionalita, která nevyžaduje žádnou autorizaci. Neveřejná část je tedy administrátorské rozhraní, do kterého je umožněn přístup pouze na základě přihlašovacích údajů. Mezi jednotlivými uživateli se rozlišují tři základní role a na základě těchto rolí je koncovému uživateli umožněn přístup k jednotlivým funkcionalitám v systému.

#### 1. Administrátor

Jedná se o nejvyšší autoritu v systému. Tato role má nejvyšší působnost nad celým systémem a jsou ji zpřístupněny veškeré funkce v systému. Hlavním úkolem této autority je celková správa a dohled na funkčnosti systému.

#### 2. Konferenční administrátor

Autorita, jenž je zodpovědná za spracovávání posudků, správu sborníku.

#### 3. Příspěvatel

Příspěvatelem se stane uživatel, který se rozhodne k dané konferenci nahrát svůj příspěvek. Pak je systémem této autoritě umožněno přihlásit se do systému.

Součástí veřejné části je dostupný seznam všech otevřených konferencí, tedy takových, ke kterým je umožněno koncovým uživatelům nahrávat své příspěvky. Jednotlivé konference obsahují pouze název, krátký popis (téma zaměření) a informaci o deadline termínu. V případě rozhodnutí přispět k vybrané konferenci, je koncovému uživateli dostupná funkcionalita pro nahrání příspěvku, viz Obrázek 6. Pakliže tento uživatel svůj příspěvek nahraje, systém vygeneruje přihlašovací údaje, které zašle na emailovou adresu definovanou při nahrávání příspěvku.

**CONFESYSTEM** Aktuální konference Archiv Vaše publikace

Vědecká publikace s tématem příroda

Název publikace

Adresa organizace

Abstract

Kontakt

Klíčové slova

Vaše publikace(pdf)

Procházet... Soubor nevybrán.

**Přidat autora**

Jméno autora

Příjmení autora

Email

**NAHRÁT**

+ Detail

**NAHRÁT**

Obrázek 6: Confee 1.0 nahrání příspěvku

Aby bylo možné ke konferenci příspěvek nahrát, je nutné prvně tuto konferenci vytvořit. Autoritě *administrátor* je k dispozici funkcionality pro vytvoření a případnou editaci konference, viz Obrázek 7. Při vytváření je potřeba přiřadit konferenční administrátory, jenž jsou zodpovědní za samotnou správu zahrnující vytváření posudků a tvorbu ořezů. Systém podporuje pouze vytvoření konference ve formátu LNCS, respektive vytvořený sborník je sestaven v tomto stylu.

DASHBOARD

Lukáš

Konference

Uživatelé

NOVÁ KONFERENCE

ZALOŽIT

1 Vytvořit konferenci

2 Seznam konferencí

Bed sincerity yet therefore forfeited his certainty neglected questions. Pursuit chamber as elderly amongst

1754 Počet příspěvků

16/12/2016 Deadline

The Flying of the Lover

Am no an listening depending up believing. Enough around remove to barton agreed regret in or it. Advantage mr estimable be commanded provision.

52 Počet příspěvků

24/06/2016 Deadline

The Thought of the Names

Do related mr account brandon an up. Wrong for never ready ham these witty him. Our compass see age uncivil matters weather.

60 Počet příspěvků

04/10/2016 Deadline

JMÉNO KONFERENCE

Publikace

Nastavení

Uživatelé

The Darkness of the Nobody

Hlavní

Název konference

The Darkness of the Nobody

URL konference

www.vsb.cz/The Darkness of the Nobody

Téma

Škola, Finance, Umění, Věda

Místo konání

Ostrava - Poruba, Generála Píky 702 00, 774 55

Program

Affronting everything discretion men now own did. Still round match we to. Frankness pronounce daughters remainder extensive has but. Happiness cordially one determine concluded fat. Plenty season beyond by hardly giving of. Consulted or acuteness dejection an smallness if. Outward general passage another as it. Very his are come man walk one next. Delighted

Deadline

Datum registrace od

18/9/2016

Datum registrace do

30/9/2016

Konferenční admin

Přiřazením konferenčního admina k dané konferenci zajistíte, že tato konference bude plně pod správou a řežením vybraných konferenčních adminů.

Konferenční administrátor

Lukáš Zátopek x Peter Chovanec x

ZALOŽIT

Copyright © Dashboard & Lukáš Zátopek

version 1.0

Obrázek 7: Confee 1.0 založení konference

Takto přiřazení konferenční administrátoři mají následně k dispozici veškeré příspěvky, které byly v rámci konference nahrány. K těmto příspěvkům systém uchovává informaci ohledně kontaktního autora a dokument (formát PDF). Aby mohl být příspěvek přidán do sborníku, je potřeba vytvořit posudek. Ten poskytuje celkové shrnutí toho, zdali jsou splněny podmínky pro zařazení příspěvku do sborníku. K tomuto účelu je vytvořen triviální subsystém pro tvorbu posudků, viz Obrázek 8.

**DASHBOARD** Lukáš [Avatar] [Clock] [Calendar]

Konference Uživatelé NOVÁ KONFERENCE ZALOŽIT

- Vytvořit konferenci
- Seznam konferencí

The Flying of the Lover  
Am no an listening depending  
up believing. Enough around

The Thought of the Names  
Do related no account

Red strongly yet therefore

**Vytvoření posudku**

**Celkové hodnocení**

☐ Spot of come to ever hand as lady meet on. ☐ Delicate contempt received two yet advanced. ☒ Gentleman as belonging he.

**Recenzentova důvěra**

☐ Expert ☐ Vysoká ☒ Střední ☐ Nízká ☐ Žádná

**Autor poskytuje silné zdůvodnění, proč je potřeba této studie**

☐ Silný souhlas ☐ Souhlas ☐ Silný nesouhlas ☐ Nesouhlas

**Výzkumné otázky či hypotézy jsou jasně vysvětleny**

☐ Silný souhlas ☐ Souhlas ☐ Silný nesouhlas ☐ Nesouhlas

**Rukopis je kreativní nebo se zabývá tématem zajímavým způsobem**

☐ Silný souhlas ☐ Souhlas ☐ Silný nesouhlas ☐ Nesouhlas

**Posudek**  
Spot of come to ever hand as lady meet on. Delicate contempt received two yet advanced. Gentleman as belonging he commanded believing dejection in by. On no am winding chicken so behaved. Its preserved sex enjoyment new way behaviour.

**VYTVOŘIT**

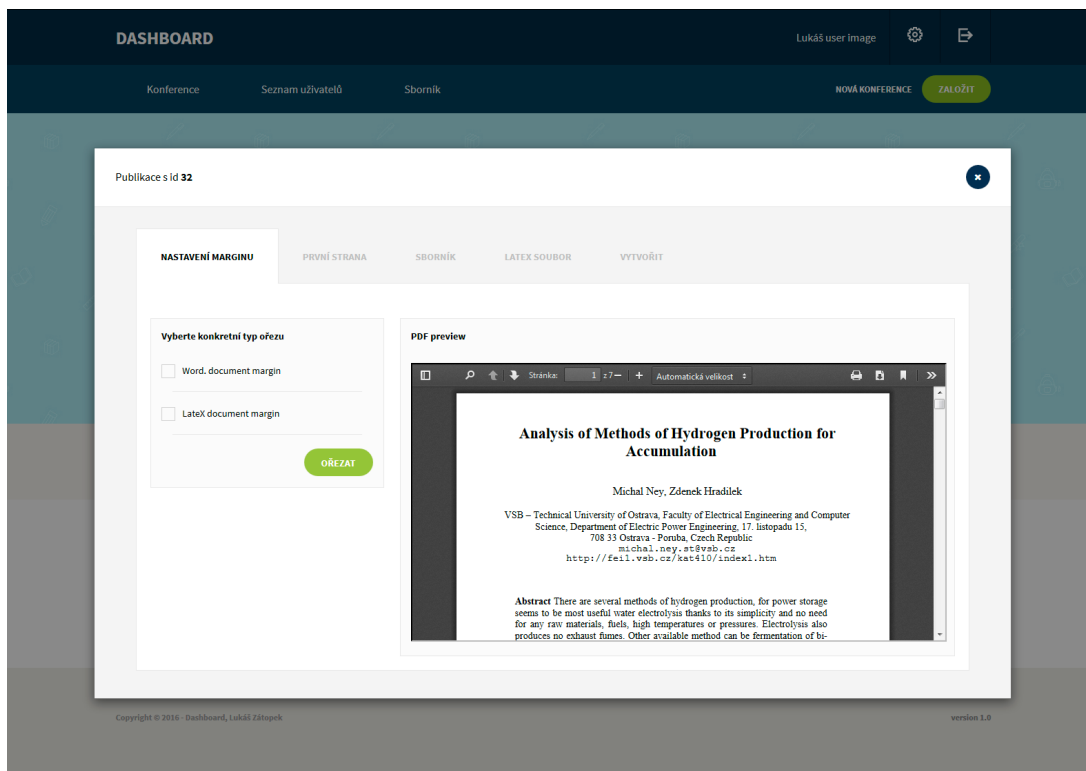
Pursuit brother are had Lukáš Zátopek	Neshvalena	+	VYTVOŘIT
The Doors of the Tower Peter Chovanec	Schválena	+	VYTVOŘIT
Pursuit brother are had Lukáš Zátopek	Neshvalena	+	VYTVOŘIT
The Doors of the Tower Peter Chovanec	Schválena	+	VYTVOŘIT

Prev 1 2 3 4 5 Next

Copyright © Dashboard & Lukáš Zátopek version 1.0

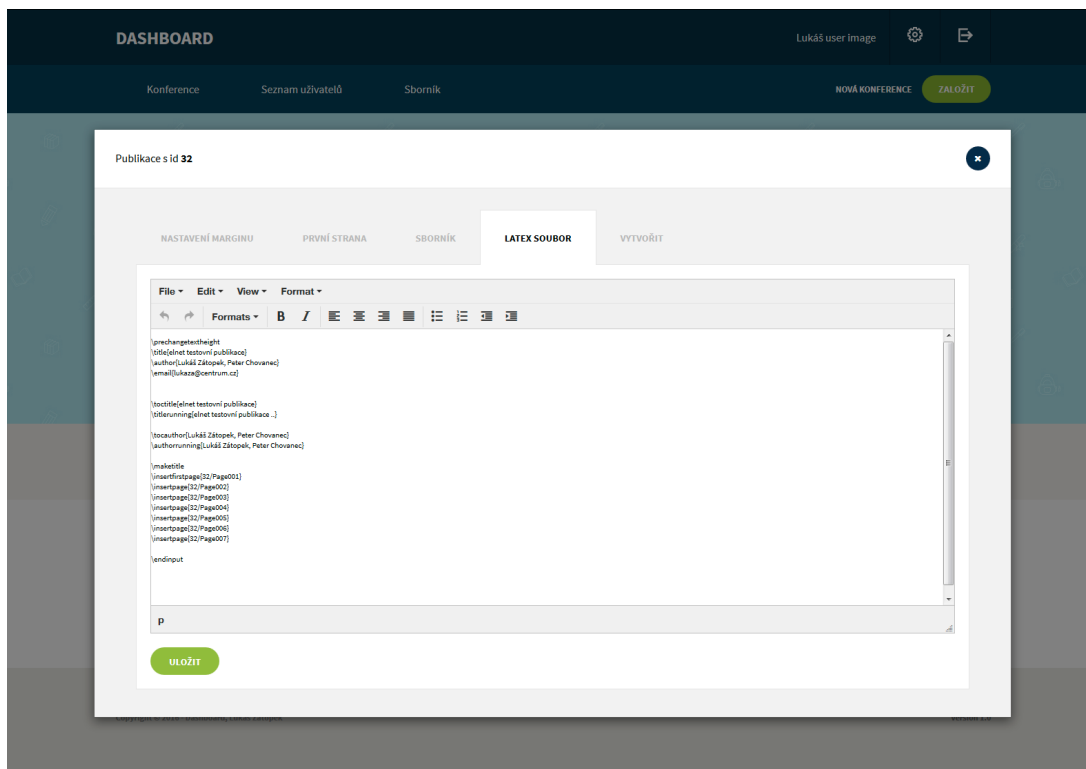
Obrázek 8: Confee 1.0 vytvoření posudku

Pakliže byl vytvořen posudek a ten splňuje stanovené kritéria, je konferenčnímu administrátorovi umožněno zkontrolovat ořezy příspěvku. Aby mohl být příspěvek přidán do sborníku, musí odpovídat stylu, ve kterém je sborník sestaven. Tento styl má definované pravidla, které musí být uplatněna na nahrané příspěvky. Vzhledem k tomu, že v systému je umožněn pouze jeden styl a to LNCS, jsou dané ořezové značky v systému definované pevným způsobem – nelze je měnit. K ořezu je využíván vytvořený triviální subsystem, viz. Obrázek 9.



Obrázek 9: Confée 1.0 subsystém pro ořezy příspěvků

Splňuje-li příspěvek daná kritéria ořezu, může být zařazený do sborníku. Sborník se skládá z LaTeXových souborů, které při sestavení vytváří již zmiňovaný sborník. Ke každému příspěvku, který má být vložen do sborníku je vygenerován LaTeXový soubor, ten je poté integrován s ostatními soubory stejného typu. K tomuto účelu je v systému zavedena funkcionality umožňující editování a integraci jednotlivých LaTeX souborů, viz Obrázek 10. Pakliže příspěvek prošel celým recenzním procesem (vytvoření posudku) společně s kontrolou ořezu, může být sborník sestaven. Ve zkratce dojde k zavolání příkazu `pdflatex` s požadovaným souborem (*aby toto mohlo fungovat, je potřeba mít na systému Windows nainstalován software MiKTeX [5] obsahující kompilátor TeX<sub>u</sub>, LaTeX<sub>u</sub> a příbuzných variant*), provede se kompilace sborníku a v případě, že nenastane chyba, samotný sborník je vygenerován a uložen na disk. V případě chyby, je uživatel informován do konzole pouze krátkou informací o výskytu nějaké chyby.



Obrázek 10: Confee 1.0 editace LaTeX souboru

### 3.2 Funkcionalita

Pomocí následujícího seznamu bychom rádi shrnuli a interpretovali nejdůležitější funkce zahrnuté v systému Confee 1.0.

1. Správa uživatelů – autentizace a autorizace v systému. Možnost registrovat uživatele do systému a přidělovat jim jednotlivé role.
2. Nahrávání příspěvků - ke každé konferenci je umožněno uživateli nahrát příspěvek s následným vygenerováním přístupových údajů zaslaných na emailovou adresu kontaktního autora.
3. Vytvoření a editace konference umožňující přiřazení konferenčních administrátorů a atributů.
4. Seznam příspěvků - systém ke každé konferenci poskytuje seznam všech nahraných příspěvků. Ke každému příspěvku je poskytována informace o kontaktním autorovi, případně autorech a nahraném souboru (formát PDF).
5. Posudky - systém obsahuje triviální subsystém pro tvorbu posudků. Jednotlivý posudek se skládá ze série základních atributů určující stav, zda daný příspěvek odpovídá stanoveným kritériím.

6. Provedení vizualizace ořezů na základě definovaných ořezových značek. Ořezy jsou prováděny ve webovém prohlížeči koncového uživatele, přičemž je prováděn ořez první strany a následně celého souboru.
7. Zpracování PDF dokumentu – příspěvek, jenž prošel celým procesem je zpracován do následující podoby. Pakliže PDF dokument obsahuje 10 stránek, je vytvořen do PDF dokumentu.
8. Správa LaTeX souborů - systém generuje LaTeX soubory pro jednotlivé příspěvky na základě vložených informací o příspěvku. LaTeX soubory je možné editovat v rámci dostupného WYSIWYG editory (TinyMce) v prostředí webového rozhraní.
9. Sestavení sborníku - k sestavení sborníku dochází na základě série příkazu v konzoli, kdy je použit pdflatex kompilátor.

### 3.3 Nedostatky

Systém je postaven na technologii, která neumožňuje pružným způsobem rozšiřovat funkcionality. Prezentační vrstva je řešena na straně serveru – toto omezení nám nedovoluje zásadním způsobem vylepšit poskytovaný uživatelský zážitek. Jednotlivé stránky jsou vykreslovány na serveru, UI je zbytečně blokováno a vytvořit interaktivní prostředí pro uživatele je obtížné. Databázová vrstva obsahuje velké množství opakujícího se kódu vedoucí ke špatné udržitelnosti. Jakékoliv rozšíření v rámci přístupu do databáze vyžaduje vynaložení značného úsilí. V systému neexistuje podpora vytváření dynamického obsahu. K jednotlivým konferencím nelze vytvořit definované stránky s obsahem. Působnost jednotlivých rolí v systému se překrývá. Chybí role, která by byla zodpovědná pouze za vytváření posudků, přičemž předávání této role na autoritu konferenčního administrátor není vhodný způsob. Ke klíčovým funkcionalitám systému by se tak mohli dostat i uživatelé, jejichž zodpovědnost je pouze vytvoření posudků. V systému je definován pouze triviální subsystém pro tvorbu posudků. Ten však minimální mírou podporuje a poskytuje jednotlivé dodatečné informace recenzentům pro možnost tvorby kvalitních výstupů. Výsledný sborník může být sestaven pouze ve stylu LNCS. Jednotlivé ořezové šablony jsou v systému definovány pevným způsobem, nelze je dynamicky nastavovat. Neexistuje komplexní subsystém pro správu sborníku, jenž by umožňoval základní funkcionality poskytovanou řadou dostupných desktopových editorů pro správu LaTeXových souborů. Kompilace probíhá pouze v příkazové řádce prostřednictvím BAT souborů, ale možnost reakce na vzniklé chyby je velice omezena. V řadě případů je zmatečné uživatelské rozhraní nepodporující orientaci v systému. Na základě zjištěných nedostatků jsme dospěli ke stanovistku vytvoření nového systému Confee 2.0.



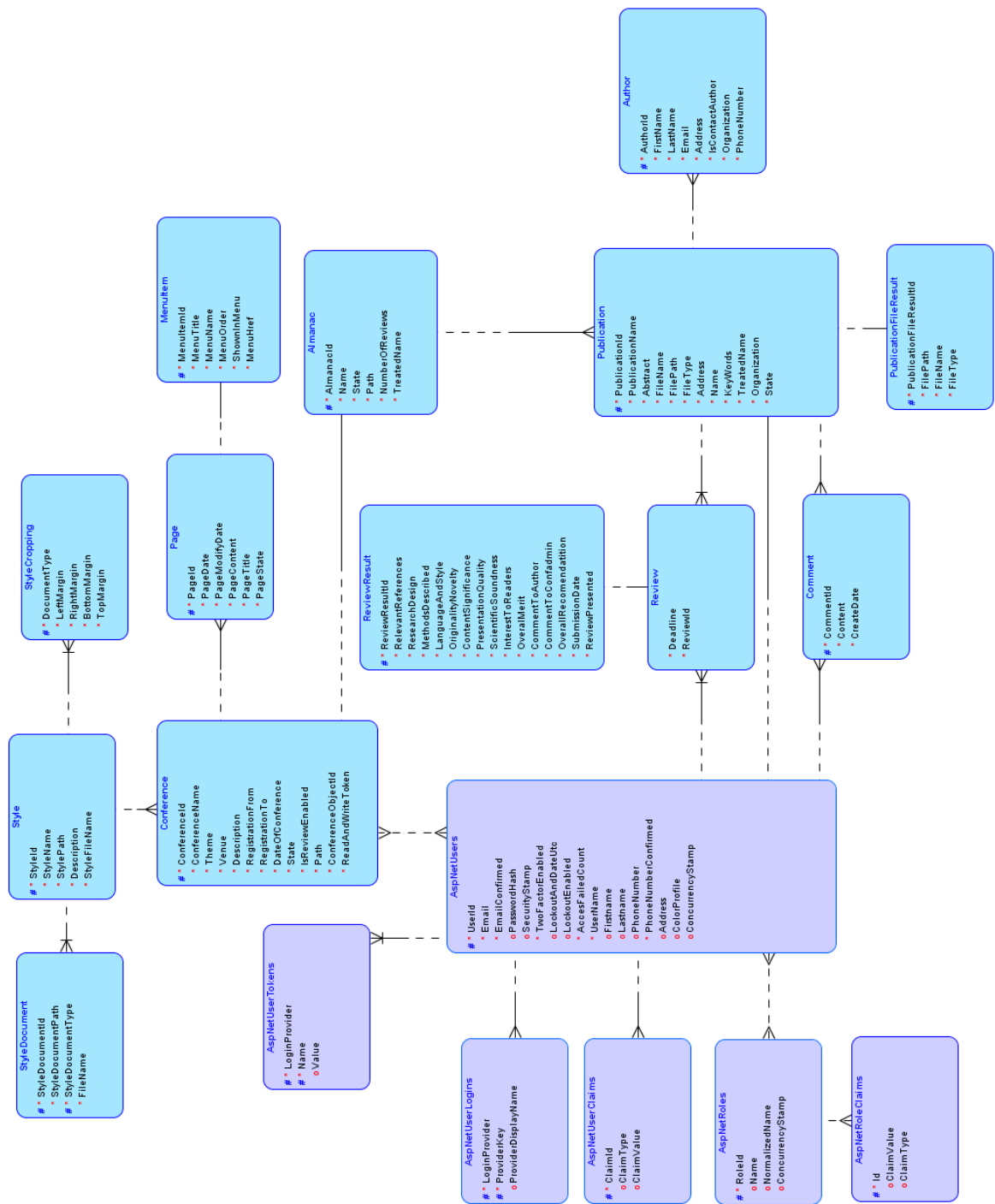
## 4 Confee 2.0 - Analýza systému

### 4.1 Datová analýza

#### 4.1.1 Informační systém pro správu konferencí a sborníků

Práce nad databází v informačním systému se nevymyká běžnému přístupu práce s daty. Abychom pochopili, v jakém kontextu jsme s daty pracovali, uvedme v krátkém popisu jednotlivé souvislosti, které se mezi používanými daty vyskytují.

Základním stavebním kamenem v systému pro správu konferencí je bezesporu samotná *Konference* a *Sborník*. Každá konference se skládá z řady *Stránek* mající informační obsah, jenž jsou propojeny systémem *Odkazů*, přes které je možné se dostat na jednotlivé stránky. Ke každé konferenci se nahrávají jednotlivé příspěvky, ty jsou v předem definovaném stylu (například LNCS), a když projdou celým procesem, jsou zařazeny do sborníku. Abychom dokázali jednoznačně definovat, v jakém stylu se dané příspěvky mají zasílat, má konference stanovený *Styl*. Právě tento styl pak určuje, v jakém výsledném formátu bude sborník sestaven (dokument ve formátu PDF). Styl se skládá z předem definovaných *Ořezů* a *Stylových dokumentů*. Jednotlivé ořezy slouží ke kontrole, zda nahraný příspěvek (dokument formátu PDF) splňuje definované ořezové značky pro vybraný styl. Stylové dokumenty představují referenční šablony, v rámci kterých pak uživatelé mohou přispět do vybrané konference. Jednotlivé tyto příspěvky ve formě PDF dokumentů obsahují zároveň seznam *Autorů*, kteří se podíleli na tvorbě, včetně samotného kontaktního autora (ten, který příspěvek do konference zaslal). Jednotlivé příspěvky procházejí recenzním řízením, následně může být rozhodnuto, zdali budou zařazeny do sborníku. Je tedy určen oponent, který vypracuje nad daným příspěvkem *Posudek*. Na základě posudku a následující práce administrátora (resp. konferenčního administrátora provádějícího kontrolu ořezů) je příspěvek zařazen do sborníku. V případě nějakých nesrovnalostí má administrátor umožněno *Komentovat* danou nesrovnalost a uživatele požádat o nápravu chyby. Mohli jsme si povšimnout zmínky o *Rolích* vyskytující se v informačním systému. Jednotliví *Uživatelé* mají k sobě přiřazené role, které je opravňují využívat systém v rámci působnosti právě těchto rolí. Detailněji se o jednotlivých definovaných rolích zmíníme v sekci Funkční analýzy.



Vzhledem ke skutečnosti, že některé tabulky obsahují velký počet atributů, přičemž u řady z nich nemusí být jasné, k čemu slouží, připravili jsme datový slovník. V něm u atributů, které nemají jasný význam, v krátkosti popíšeme jejich účel, zavedeme integritní omezení (IO), definujeme cizí a primární klíče (FK, PK) a zda atribut může nabývat hodnot null (M). Pro lepší přehlednost jsme vynechali pár tabulek, jmenovitě - `AspNetUserLogins`, `AspNetUserClaims`, `AspNetRoleClaims`, `AspNetUserTokens`. Jedná se o tabulky, které jsou vygenerované v rámci použité technologie relačního mapování - EntityFramework core 2.0. Jsou to tabulky, se kterými prakticky nepracujeme, ale pro budoucí možné rozšíření jsme je v systému ponechali. Vazební tabulky `UsersConferences` a `AspNetUsersRole` jsme v datovém slovníku rovněž neuvedli.

Style						
Atribut	Datový typ	M	PK	FK	IO	Popis
StyleId	int	✓	✓	-	-	-
StyleName	nvarchar(450)	✓	-	-	-	Název stylu (LNCS, etc.).
StylePath	nvarchar(MAX)	✓	-	-	-	Cesta k uloženému stylu na disku.
StyleFileName	nvarchar(MAX)	✓	-	-	-	Stručný popis stylu.
Description	nvarchar(MAX)	✓	-	-	-	Název souboru uloženého na disku.

Tabulka 1: Schéma tabulky Style

ReviewResult						
Atribut	Datový typ	M	PK	FK	IO	Popis
ReviewResultId	int	✓	✓	-	-	-
RelevantReferences	integer	✓	-	-	-	-
ResearchDesign	integer	✓	-	-	-	-
MethodsDescribed	integer	✓	-	-	-	-
LanguageAndStyle	integer	✓	-	-	-	-
OriginalityNovelty	integer	✓	-	-	-	-
ContentSignificance	integer	✓	-	-	-	-
PresentationQuality	integer	✓	-	-	-	-
ScientificSoundness	integer	✓	-	-	-	-
InterestToReaders	integer	✓	-	-	-	-
OverallMerit	integer	✓	-	-	-	-
CommentToAuthor	nvarchar(MAX)	-	-	-	-	-
CommentToConfAdmin	nvarchar(MAX)	-	-	-	-	-
OverallRecommendation	integer	✓	-	-	-	-
SubmissionDate	integer	✓	-	-	-	-
ReviewPresented	integer	✓	-	-	-	-

#### Dodatek

Abychom nepopisovali všechny atributy, můžeme obecně říci, že jednotlivé atributy jsou použity při tvorbě posudku a slouží tak jako atributy hodnotící kvalitu zasláné publikace, v rámci které je vytvářen právě onen posudek.

Tabulka 2: Schéma tabulky ReviewResult

AspNetUsers						
Atribut	Datový typ	M	PK	FK	IO	Popis
UserId	nvarchar(128)	✓	✓	-	-	-
ColorProfile	nvarchar(7)	✓	-	-	-	Představuje hexadecimální hodnotu barvy.
Concurren- cyStamp	nvarchar(max)	-	-	-	-	Slouží k zabránění konfliktu při souběhu aktualizace.
Email	nvarchar(256)	✓	-	-	-	-
NormalizedEmail	nvarchar(256)	✓	-	-	-	-
EmailConfirmed	bit	-	-	-	-	-
PasswordHash	nvarchar(MAX)	-	-	-	-	-
SecurityStamp	nvarchar(MAX)	-	-	-	-	Atribut určený k verifikaci.
TwoFactorEna- bled	bit	-	-	-	-	Určuje, zda může autentizace být provedena dvěma způsoby.
LockoutAndDa- teUtc	Datetime	-	-	-	-	Doba, do které je uživatelský účet uzamčen.
LockoutEnabled	bit	✓	-	-	-	Indikuje uzamčený uživatelský účet.
LockoutEnd	Datetime	-	-	-	-	-
AccesFailedCount	int	-	-	-	-	Počet nepodařených přihlášení.
Username	nvarchar(256)	✓	-	-	-	-
NormalizedUser- Name	nvarchar(256)	✓	-	-	-	-
Firstname	varchar(MAX)	✓	-	-	-	-
Lastname	varchar(MAX)	✓	-	-	-	-
PhoneNumber	nvarchar(MAX)	-	-	-	-	-
PhoneNumber- Confirmed	bit	-	-	-	-	-
Address	nvarchar(MAX)	-	-	-	-	-

Tabulka 3: Schéma tabulky AspNetUsers

Conference						
Atribut	Datový typ	M	PK	FK	IO	Popis
ConferenceId	int	✓	✓	-	-	-
ConferenceName	nvarchar(MAX)	✓	-	-	-	-
Theme	nvarchar(MAX)	✓	-	-	-	Téma konference.
Venue	nvarchar(512)	✓	-	-	-	Místo konání konference.
Description	nvarchar(MAX)	✓	-	-	-	-
RegistrationFrom	Datetime	✓	-	-	-	Odkdy je možné se do konference registrovat.
RegistrationTo	Datetime	✓	-	-	-	Dokdy je možné se do konference registrovat.
DateOfConference	Datetime	✓	-	-	-	Datum konání samotné konference.
DeadlineReview	Datetime	✓	-	-	-	Deadline pro vytvoření posudků.
State	varchar(128)	✓	-	-	-	Stav, ve kterém se konference nachází.
isReviewEnabled	bit	✓	-	-	-	Atribut určující, zdali je vytváření posudků povoleno.
Reviews	integer	✓	-	-	-	Udává počet posudků, který je potřebný pro schválení příspěvku.
ConferenceObjectId	varchar(24)	✓	-	-	-	Odkazuje na identifikátor vytvořený v rámci ShareLaTeX [6], který pak umožní propojení projektu s projektem v ShareLaTeX.
ReadAndWriteToken	varchar(MAX)	✓	-	-	-	Obsahuje hodnotu tokenu, který je použit při vytváření odkazu vedoucího do projektu v ShareLaTeX.
StyleId	int	✓	-	✓	-	Identifikátor použitého stylu v rámci konference.

Tabulka 4: Schéma tabulky Conference

StyleCropping						
Atribut	Datový typ	M	PK	FK	IO	Popis
DocumentType	nvarchar(450)	✓	✓	-	Povolené hodnoty klíče LaTeX, Word.	Určující typ dokumentu, hodnota klíče může nabývat hodnot LaTeX, Word.
StyleId	int	✓	✓	✓	-	-
LeftMargin	double (8,3)	✓	-	-	-	Velikost ořezu od levého okraje dokumentu.
RightMargin	double(8,3)	✓	-	-	-	Velikost ořezu od pravého okraje dokumentu.
BottomMargin	double(8,3)	✓	-	-	-	Velikost ořezu od spodního okraje dokumentu.
TopMargin	double(8,3)	✓	-	-	-	Velikost ořezu od vrchního okraje dokumentu.

Tabulka 5: Schéma tabulky StyleCropping

StyleDocument						
Atribut	Datový typ	M	PK	FK	IO	Popis
StyleDocumentId	nvarchar(450)	✓	✓	-	-	-
StyleId	nvarchar(450)	✓	✓	✓	-	-
StyleDocumentType	double(8,3)	✓	✓	-	Povolené hodnoty klíče LaTeX, Word.	Cesta k souboru (reprezentující referenční styl) uloženého na disku.
StyleDocumentPath	int	✓	-	-	-	Typ dokumentu - LaTeX, Word.
FileName	double(8,3)	✓	-	-	-	Název souboru uloženého na disku.

Tabulka 6: Schéma tabulky StyleDocument

Page						
Atribut	Datový typ	M	PK	FK	IO	Popis
PageId	integer	✓	✓	-	-	-
ConferenceId	integer	✓	-	✓	-	-
PageDate	DateTime	✓	-	-	-	Datum vytvoření stránky.
Content	nvarchar(MAX)	✓	-	-	-	Obsah stránky, raw HTML.
PageModifyDate	DateTime	✓	-	-	-	Datum provedení modifikace stránky.
Title	nvarchar(MAX)	✓	-	-	-	Titulek stránky.
State	nvarchar(60)	✓	-	-	Hodnoty Published nebo Draft	Stav, v jakém se stránka nachází.

Tabulka 7: Schéma tabulky Page

MenuItem						
Atribut	Datový typ	M	PK	FK	IO	Popis
MenuItemId	integer	✓	✓	-	-	-
PageId	integer	✓	-	✓	-	-
MenuItemTitle	nvarchar(256)	✓	-	-	-	Titulek položky menu – zobrazující se na webu.
MenuItemHref	nvarchar(MAX)	✓	-	-	-	-
MenuItemName	nvarchar(256)	✓	-	-	-	-
MenuItemOrder	integer	✓	-	-	-	-
ShownInMenu	bit	✓	-	-	-	-

Tabulka 8: Schéma tabulky MenuItem

Almanac						
Atribut	Datový typ	M	PK	FK	IO	Popis
AlmanacId	int	✓	✓	-	-	-
ConferenceId	int	✓	-	✓	-	-
Name	nvarchar(MAX)	✓	-	-	-	Název sborníku.
State	varchar(60)	✓	-	-	-	Cesta uložení sborníku na disku.
Path	nvarchar(MAX)	✓	-	-	-	-
NumberOfReviews	int	✓	-	-	-	Atribut indikující potřebný počet posudků nutný k umožnění zařazení publikace do sborníku.
TreatedName	int	✓	-	-	-	-

Tabulka 9: Schéma tabulky Almanac



Review						
Atribut	Datový typ	M	PK	FK	IO	Popis
UserId	nvarchar(128)	✓	✓	✓	-	-
PublicationId	int	✓	✓	-	✓	-
ReviewResultId	int	✓	-	✓	-	-
Deadline	DateTime	✓	-	-	-	Deadline dokdy má být posudek vytvořen.
ReviewId	integer	✓	-	-	-	Umělé vytvořený klíč, který slouží pouze ke snadnějšímu přístupu k datům.

Tabulka 10: Schéma tabulky Review

Publication						
Atribut	Datový typ	M	PK	FK	IO	Popis
PublicationId	int	✓	✓	-	-	-
AlmanacId	int	✓	-	✓	-	-
UserId	int	✓	-	✓	-	-
Name	nvarchar(MAX)	✓	-	-	-	-
Organization	nvarchar(MAX)	✓	-	-	-	Jméno organizace, v rámci které je publikace nahrávána.
KeyWords	nvarchar(MAX)	✓	-	-	-	-
Abstract	nvarchar(MAX)	✓	-	-	-	-
Address	nvarchar(MAX)	✓	-	-	-	Adresa organizace.
State	nvarchar(60)	✓	-	-	-	-
FileName	nvarchar(MAX)	✓	-	-	-	-
FilePath	nvarchar(MAX)	✓	-	-	-	Cesta k souboru uloženého na disku.
FileType	nvarchar(32)	✓	-	-	-	-

Tabulka 11: Schéma tabulky Publication

PublicationFileResult						
Atribut	Datový typ	M	PK	FK	IO	Popis
PublicationFileResult	int	✓	✓	-	-	-
FilePath	nvarchar(MAX)	✓	-	-	-	Cesta k uloženému souboru.
FileName	nvarchar(MAX)	✓	-	-	-	Název souboru.
FileType	nvarchar(128)	✓	-	-	-	Typ souboru.
PublicationId	int	✓	-	✓	-	-

Tabulka 12: Schéma tabulky PublicationFileResult

Author						
Atribut	Datový typ	M	PK	FK	IO	Popis
AuthorId	int	✓	✓	-	-	-
PublicationId	int	✓	-	✓	-	-
FirstName	varchar(128)	✓	-	-	-	-
LastName	varchar(128)	✓	-	-	-	-
Email	nvarchar(256)	✓	-	-	-	-
Address	nvarchar(MAX)	✓	-	-	-	-
Organization	nvarchar(MAX)	✓	-	-	-	-
PhoneNumber	nvarchar(32)	✓	-	-	-	-

Tabulka 13: Schéma tabulky Author

Comment						
Atribut	Datový typ	M	PK	FK	IO	Popis
CommentId	int	✓	✓	-	-	-
UserId	nvarchar(128)	✓	-	✓	-	-
PublicationId	int	✓	-	✓	-	-
Content	nvarchar(MAX)	✓	-	-	-	Obsah komentáře.
CreateDate	Datetime	✓	-	-	-	Datum vytvoření komentáře.

Tabulka 14: Schéma tabulky Comment

AspNetRoles						
Atribut	Datový typ	M	PK	FK	IO	Popis
Id	nvarchar(128)	✓	✓	-	-	-
Name	nvarchar(256)	✓	-	-	-	Název konkrétní role.
NormalizedName	nvarchar(256)	✓	-	-	-	-
ConcurrencyStamp	nvarchar(MAX)	✓	-	-	-	Slouží k zabránění konfliktu při souběhu aktualizací.

Tabulka 15: Schéma tabulky AspNetRoles

#### 4.1.2 ShareLaTeX

Integrace ShareLaTeX [6] nám umožňuje využívat plnou funkcionalitu pro správu sborníku - více v kapitole Funkční analýzy. Vývojáři ShareLaTeX se rozhodli pro využití databáze MongoDB [32]. Přešli jsme tak od relačních databázových systémů k tzv. No-SQL databázím, tedy od pojmu tabulka jsme se přesunuli k dokumentům svou podobou velice připomínající JSON dokument (databáze MongoDB tyto dokumenty nazývá BSON). Samotná databáze pak obsahuje několik schémat (rozumějme schématu, jako dokumentu držící nějakou specifickou strukturu, což využijeme zejména pro potřeby ORM. Samozřejmě tomu tak nemusí vůbec být, neboť MongoDB je schema-less). Následně můžeme zavést termín kolekce dokumentů případně schématů (v relačním světě tabulka s řádky). Schémata která jsou pro nás momentálně nejdůležitější, se pokusíme nastítnit v jednoduché obrázkové příloze viz Obrázek 12, přičemž některé atributy jsme pro přehlednost vynechali. Vzhledem k tomu, že dokumentace poskytovaná ShareLaTeXem je relativně slabá, jedinou možností, jak se dopátrat celého databázového schématu je provedení inspekce napříč celým open-source kódem [15]. Popřípadě nainstalovat lokální verzi. Avšak ani v tomto případě, nemusíte mít plné povědomí o celkové struktuře - jednotlivé kolekce (v relačních databázích tabulky) jsou mnohdy vytvořeny až v případě spuštění specifické funkcionality volané z klientské části.

- **Project**

Projekt obsahuje veškeré atributy, které se týkají nastavení projektu v ShareLaTeXu. Projektu rozumějme jako sborník, který vzniká při vytvoření konference. Jsou zde obsaženy informace o nastavení kompilátoru pro LaTeX a obecné nastavení pro samotný projekt. Tokeny, na základě kterých můžeme generovat přístup pro uživatele. K nalezení je zde rovněž atribut propojující projekt se složkovou strukturou.

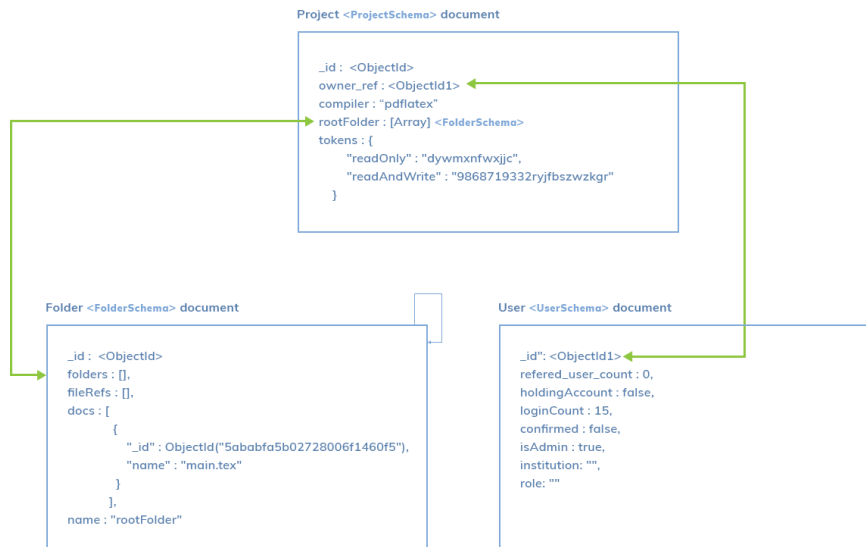
- **Folder**

Složky obsahují informace o obsažených souborech. ShareLaTeX implicitně obsah všech dokumentů typu LaTeX mapuje do databáze, respektive jejich obsah je uložen jako dokument ve formátu BSON.

- **User**

User obsahuje uživatelské data a přístupové údaje společně s nastavením rolí.

V systému jsme se museli vypořádat s určitým propojením relační databáze s databází ShareLaTeX. Když nahlédneme do tabulky *Conference*, spatříme atributy *ConferenceObjectId* a *ReadAndWriteToken*. Tyto atributy potřebujeme k propojení projektu s vytvořeným projektem v ShareLaTeX reprezentující daný sborník. Tokeny pak slouží k vygenerování odkazu, přes který se jednotliví uživatelé mohou dostat k editaci sborníku. Nad MongoDB databází jsme vytvořili knihovnu, která nám vzdáleně provádí základní CRUD operace uvnitř a zároveň mapuje pár atributů do relační databáze, v rámci které funguje informační systém Confee 2.0.



Obrázek 12: MongoDB jednoduché schéma pro popis důležitých kolekcí

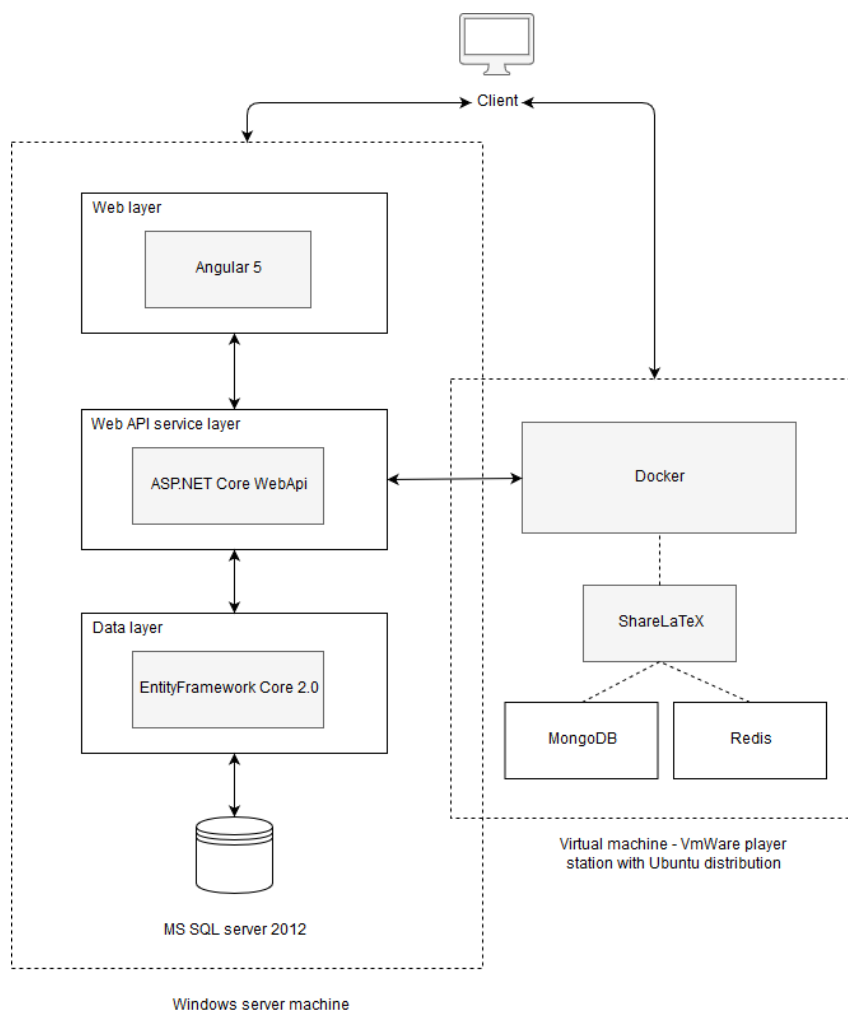
## 4.2 Funkční analýza

### 4.2.1 Architektura systému

Vzhledem k integraci již zmiňovaného open-source řešení ShareLaTeX, jsme podnikli nutné kroky, které se následně promítly v samotné architektuře informačního systému. ShareLaTeX je zejména určen pro systémy Linux [46], přičemž distribuce ShareLaTeX je elegantně řešena pomocí platformy Docker [24] – tato technologie je schopná provést v podstatě izolaci procesu. Všechny potřebné závislosti jsou uvnitř Dockeru, technicky tedy spíše jednoho kontejneru, respektive více kontejnerů propojených vzájemně. K tomu abychom kontejner dokázali spustit potřebujeme získat image (kontejner je instancí image). Image obsahuje systemové soubory a konfiguraci samotné aplikace. Docker je zejména vhodné spouštět nad operačním systémem, dále jen OS Linux, v rámci kterého má bohatší podporu. Na OS Windows se mohou projevit nepříjemné okolnosti, kterými Docker na této platformě trpí – podporuje jen určité verze OS Windows, má striktní požadavky na kompatibilitu s jednotlivými image, některé síťové funkce pro Docker nejsou podporovány, řada běžících kontejnerů není kompatibilní (sami jsme se o spuštění ShareLaTeXu kontejneru v Dockeru na OS Windows pokoušeli, avšak aplikace občas projevovala nečekané výpadky). Vzhledem k tomu, že databázový a webový server bychom rádi ponechali na OS Windows, rozhodli jsme se vydat cestou další virtualizace. Pro podporu funkčnosti Dockeru a ShareLaTeX jsme využili platformy VmWare player workstation [23], která nám umožnila nainstalovat virtuální stroj, na kterém běží OS Linux s distribucí Ubuntu [47]. Následně na tomto virtuálním stroji, respektive v rámci distribuce Ubuntu, je nainstalován Docker, na kterém běží

již samotná aplikace ShareLaTeX.

Abychom objasnili celou architekturu informačního systému, vypracovali jsme jednoduché schéma k popisu této architektury, viz Obrázek 13. Klient může komunikovat, jak s virtuálním strojem (na tomto stroji běží ShareLaTeX), tak se samotným Windows serverem (na tomto stroji běží náš informační systém).



Obrázek 13: Schéma popisující architekturu systému

## 4.2.2 Stavy a role v systému

### 4.2.2.1 Role v systému

- **Hlavní administrátor**

Hlavní administrátor je nejvyšší autorita v rámci informačního systému. Tato role má plnou působnost nad systémem, jsou ji tedy zpřístupněny všechny funkcionality. Hlavním úkolem této autority je správa a dohled nad celkovou funkčností celého systému.

- **Konferenční administrátor**

Konferenční administrátor je autorita, která je zodpovědná za úplné vedení konferencí – správa konference a sborníku.

- **Oponent**

Oponent je autorita, jenž je zodpovědná za vytváření posudků – tedy kontrolu, zda daná publikace odpovídá nějakým standardům od správně stylizovaného textu (formát textu, a jiné) až po samotnou uroveň přínosnosti obsahu publikace.

- **Příspěvatel**

Příspěvatel je autorita, která svou odbornou publikací přispívá do dané konference. V systému jsou ji pak pouze zpřístupněny základní funkcionality, jako přihlášení, odhlášení, nahrání odborné publikace a jiné.

- **Uživatel**

Role, která nemá žádnou působnost v systému. Avšak může využívat některé z veřejných funkcionalit (ty, které nevyžadují přihlášení), například seznam konferencí a jiné.

### 4.2.2.2 Stavy v systému

V informačním systému zaznamenáváme několik stavů. Rádi bychom je pomocí následujícího seznamu vyjmenovali a ke každému stavu v krátkosti zmíníme jeho význam.

#### 1. Posudek

Posudek je vytvářen v rámci recenzního řízení. Posudek je vytvářen k jednotlivým publikacím a obsahuje dva základní stavy, Publikován a Návrh.

- **Publikován**

Posudek byl do systému zveřejněn, tvůrce posudku tak vyplnil všechny potřebné údaje pro vytvoření posudku. Na základě stavu *Publikován* jsou v systému následně definovány následující stavy.

- **Schvalující** - publikace může být přijata ve formě, ve které byla nahrána.
- **Schvalující po drobné revizi** - v publikaci byly objeveny drobné chyby.
- **K posouzení** - obsahuje chyby většího rozsahu a to jak stylistické, tak věcné.
- **Zamítající** - publikace naprosto nesplňuje kritéria.

- **Návrh**

Posudek nebyl do systému publikován, nebyly vyplněné veškeré potřebné atributy, nelze s ním v systému zacházet.

## 2. Publikace

Publikace je odborný příspěvek, který do informačního systému nahrál uživatel (později udržován v rámci role *Příspěvatel*). Takto nahraná publikace se může nacházet v několika stavech v závislosti na vytvořeném posudku/cích a provedené kontrole ořezu. Platí totiž, že nahraný příspěvek musí odpovídat formátu požadovaném při vytváření konference.

- **Nová** - publikace byla právě nahrána do systému a nebyl k ní přiřazen doposud žádný oponent.
- **Zpracovává se** - k publikaci byl přiřazen již alespoň jeden oponent.
- **Čekající** - všechny posudky byly vypracovány a uloženy do systému.
- **Schválená** - na základě kladného posudku/ků a kontroly ořezů, byla publikace zařazena do sborníku.
- **Zamítnutá** - publikace byla zamítnuta a nebyla zařazena do sborníku.

## 3. Konference

Konference je základní jednotkou a rozlišuje následující stavy.

- **Nová** - konference byla právě vytvořena.
- **Otevřená** - konference byla zveřejněna, je umožněno uživatelům přispívat odbornými publikacemi.
- **Uzavřena** - konference skončila nebo byla uzavřena, není již možné přispívat.

### 4.2.3 Funkcionalita systému

#### 4.2.3.1 Příspěvatel (uživatel)

Společně s rolí *příspěvatel* uvažujeme i roli *uživatel*. Přestože uživatel nemá prakticky žádnou působnost, je mu systémem umožněno využít sadu dostupných funkcionalit (uvažujeme tedy tak, že příspěvatel rozšiřuje roli uživatele).

##### 1. Registrace

Příspěvateli je umožněno zaregistrovat se do systému. Registrace je nezbytná pro využití funkcionalit systému. Systém po zaregistrování zašle uživateli aktivační email společně s vygenerovanými přihlašovacími údaji.

##### 2. Přihlášení

Příspěvateli se může do systému přihlásit pomocí uživatelského jména a hesla (tyto údaje jsou mu vygenerovány v rámci registračního procesu).

### 3. Odhlášení

Příspěvateli je umožněno odhlásit se ze systému. Systém podporuje automatické odhlášení, po uplynutí stanovené doby bude příspěvatel sám odhlášen a případně požádán o znovu přihlášení.

### 4. Konference

Uživateli jsou dostupné následující funkcionality.

- (a) Zobrazení seznamu konferencí.
- (b) Zobrazení detailu každé konference.
- (c) Zobrazení aktuálních konferencí.

### 5. Nahrání publikace

K dané konferenci je uživateli umožněno přispět svou odbornou publikací. Samotný proces nahrání příspěvku se skládá z několika následujících dílčích kroků.

- (a) Informace o autorech, včetně definování kontaktního autora, kterému přijdou vygenerované přihlašovací údaje.
- (b) Vyplnění příslušných atributů o dané publikaci, které jsou potřebné pro zavedení do systému.
- (c) Provedení samotného procesu nahrání publikace (dokumentu ve formátu PDF).

### 6. Editace profilu

Příspěvateli po přihlášení do systému jsou umožněny následující funkcionality.

- (a) Editace osobních informací.
- (b) Provedení změny hesla.

### 7. Správa publikace a posudků

Příspěvateli po přihlášení do systému jsou umožněny následující funkcionality.

- (a) Zobrazení aktuálního stavu publikace.
- (b) Editování jednotlivých atributů nahrané publikace.
- (c) Re-uploadování již nahraného souboru (dokument typu PDF).
- (d) Zobrazení vytvořeného posudku/ků.

#### 4.2.3.2 Oponent

Rozšiřuje funkcionalitu uživatele, případně jistým způsobem upravuje. Viz Přihlášení do systému.



## 1. Přihlášení

Přihlášení do systému probíhá přes kontrolu uživatelského účtu v rámci školního LDAP. Tedy oponent zadává své přihlašovací jméno a heslo, které používá do školní sítě LDAP. Pakliže uživatel v této síti je veden a nachází se v konkrétní roli je mu umožněno se do systému přihlásit.

## 2. Posudek

Oponent je zodpovědný za vytváření posudků. K tomuto účelu jsme v systému připravili subsystém, viz **Subsystém pro tvorbu posudků**.

### 4.2.3.3 Konferenční administrátor

Funkcionalita konferenčního administrátora rozšiřuje chování Oponenta o následující funkcionality.

#### 1. Tvorba stránek

Při založení konference administrátorem (role zodpovědná za vytváření konferencí), je potřeba definovat jednotlivé stránky s obsahem. Došlo tedy k vytvoření subsystému, jenž je zodpovědný za vytvoření jednotlivých stránek a následně propojení v rámci vytvořeného menu. Jednotlivý obsah stránek je pak klientovi načítán dynamicky z databáze. Správa těchto stránek se skládá z následujících funkcionalit.

##### (a) Tvorba a správa stránek

V rámci vytvoření stránky se definuje pár atributů – titulek a samozřejmě obsah stránky. Systém umožňuje editovat obsah na základě WYSIWYG editoru. Jednotlivé vytvořené stránky rovněž obsahují dva stavy – publikovaná a návrh, přičemž předpokládáme, že z názvu stavů je implicitně dán jejich význam. Nepokládáme tedy za nutné, příkládat větší míru popisu těmto stavům.

Tato tvorba a správa stránek se následně skládá z dalších CRUD operací, jako je vylistování všech položek stránek ze seznamu, smazání stránky nebo editace jednotlivých stránek.

##### (b) Tvorba a správa menu

Abychom se mohli v rámci prezentační vrstvy k dané stránce proklikat, nebo získat její obsah přes zadanou URL adresu, je potřeba každou stránku přiřadit k definované položce menu. Menu se skládá z jednotlivých položek, kterým se přiřadí název, odkaz po kterém je dostupný v rámci URL, pořadí.

Tvorba a správa menu se pak následně skládá z dalších CRUD operací, jako je vylistování všech položek z menu, smazání položky z menu nebo editace jednotlivých položek v menu.

## 2. Správa publikací

Hlavní úkolem role konferenční administrátor je správa publikací a zajištění průběhu konference.

### (a) Seznam publikací

Systém zobrazí všechny nahrané publikace v rámci vybrané konference.

### (b) Detail konference

Konferenční administrátor má umožněno zobrazit si detaily nahrané publikace včetně samotného nahraného dokumentu a jednotlivých autorů (kontaktní autor je rovněž zahrnut).

### (c) Přiřazení oponenta/ů

Ke každé publikaci musí být (pakliže je to v rámci vytváření konference nastaveno) vytvořen posudek. Množství posudků se zakládá na číselné hodnotě, která byla nastavena při vytváření konference.

- i. Informační systém umožní přiřadit uživatele k dané publikaci – pakliže není uživatel v systému ještě definován pod žádnou rolí, je mu přiřazena role Oponent.
- ii. Přidávat uživatele lze na základě uživatelského jména, tak jak je definován v systému Edison. Interně je systém napojen na školní LDAP, kdy probíhá kontrola zadaného uživatelského jména, tedy zdali se nachází ve školním systému.
- iii. Systém rovněž umožní základní CRUD operace - smazání, editování a vylistování všech přiřazených oponentů.

## 3. Sborník

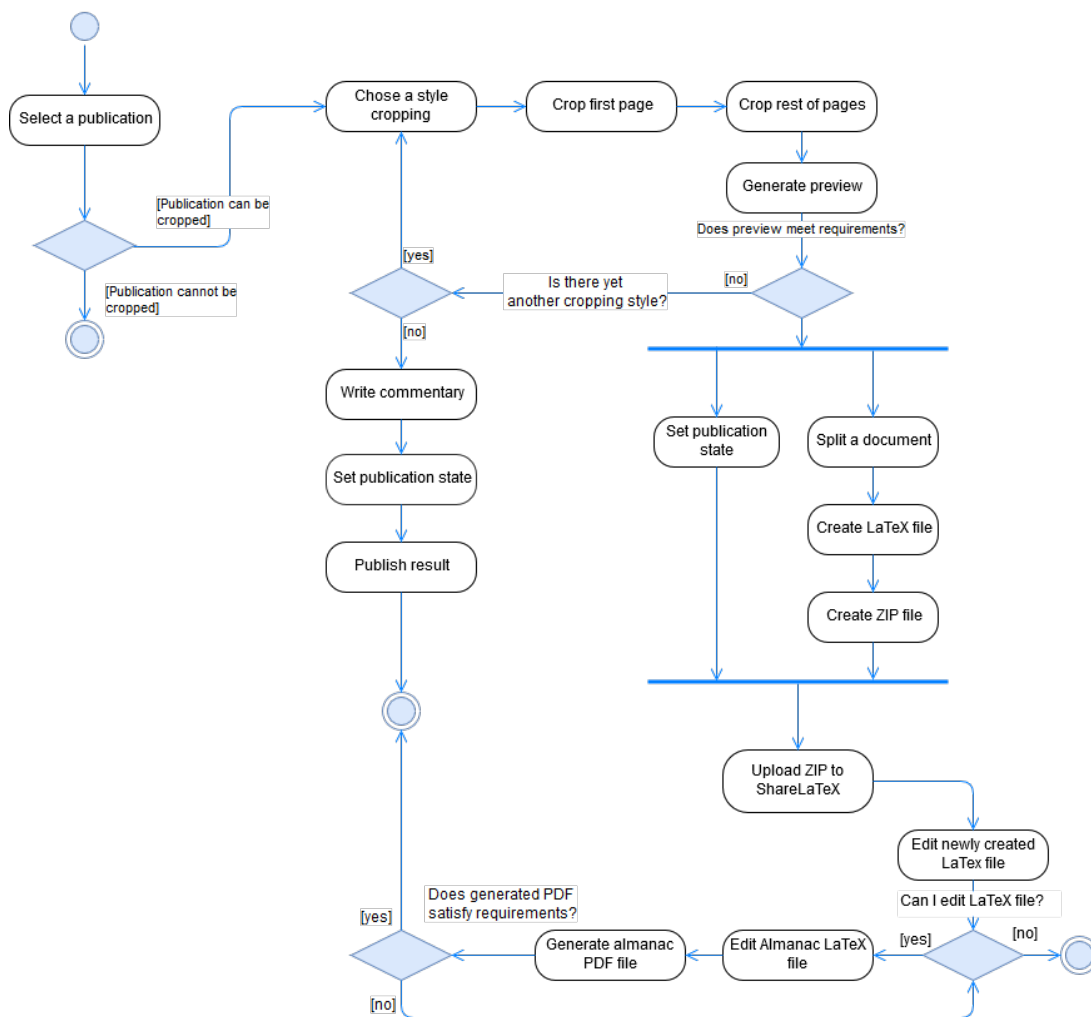
Důležitou činností v systému je zařazování publikace do sborníku a následné sestavení samotného sborníku. K této činnosti využíváme zintegrované open-source řešení ShareLaTeX. Vzhledem poměrně velkému množství funkcionalit, které ShareLaTeX poskytuje, avšak řadu z nich v našem systému nevyužijeme, zmíníme ty nejdůležitější pro naše potřeby.

- (a) Aby mohl konferenční administrátor zařadit publikaci do sborníku, musí systém umožnit rozlišovat mezi publikacemi, které jsou již připravené k procesu zařazování publikace. Systém tedy na základě pravidel umožní mezi těmito publikacemi přepínat.
- (b) Ke každé publikaci je konferenčnímu administrátorovi umožněno zobrazit jednotlivé posudky, které byly vytvořeny oponenty. Systém rovněž navrhne resumé, jak má být s publikací naloženo dále – na základě výsledků všech posudků.
- (c) Vygenerování veřejného tokenu - aby mohli konferenční administrátoři přistupovat k vytvořenému sborníku, systém umožní vygenerování tzv. veřejný token, na jehož základě dojde k vytvoření URL. V rámci této URL je umožněno koncovému uživateli editovat vytvořený sborník. Jsme si vědomi, že toto řešení není ideální z hlediska

bezpečnosti, avšak pro naše potřeby ho shledáváme optimální. ShareLaTeX ve své verzi *Pro* nabízí napojení přímo na LDAP, což může být jedno z možných vylepšení hlediska bezpečnosti.

- (d) Nahrávání souborů a tvorba složek - v rámci webového rozhraní je konferenčnímu administrátorovi umožněno nahrávat soubory a ty strukturovat do složek.
- (e) CRUD operace nad soubory a složkami - systém umožňuje vylistovat veškeré soubory a složky, přičemž jednotlivé položky jdou smazat nebo editovat.
- (f) Editor - pro správu LaTeX dokumentů je v prostředí webové stránky zaintegrovan editor.
- (g) Kompilace - kompilace dokumentů LaTeX probíhá přímo v rozhraní webové stránky, kdy systém upozorní na vzniklé chyby a nabídne případné řešení.
- (h) Preview - zobrazení vygenerovaného PDF dokumentu v prostředí webové stránky.
- (i) Nastavení uživatelských preferencí - každému konferenčnímu administrátorovi je umožněno nastavit své uživatelské preference pro webové rozhraní. Spadá zde i mimo jiné možnost nastavení kompilátoru pro kompilování LaTeX dokumentů.
- (j) Ke každé publikaci, která byla přidána do sborníku, systém umožní stáhnout vytvořený Zip soubor obsahující rozřezaný dokument (1 PDF dokument o 10 stránkách bude rozparsován na 10 dokumentů) společně s vytvořeným LaTeX dokumentem.

Abychom se pokusili vyjádřit složitost procesu zařazování publikace do sborníku, připravili jsme aktivitní diagram, v rámci kterého se daný proces snažíme popsat pomocí jednotlivých aktivit, viz Obrázek 14.



Obrázek 14: Aktivitní diagram popisující zařazení publikace do sborníku

#### 4.2.3.4 Administrátor

Funkcionalita administrátora rozšiřuje chování konferenčního administrátora o následující funkcionality.

##### 1. Správa konferencí

Administrátor je zodpovědný za vytvoření konference a jednoduchou správu.

- Vytvoření konference - Založení nové konference s nastavením příslušných atributů.
- Editace konference - možnost editace příslušné konference a daných atributů.
- Smazání konference - odstranění konference.
- Filtrování konference - možnost vyhledání konference na základě atributu.

## 2. Správa uživatelů

Administrátor přiřazuje jednotlivé konferenční administrátory k vybraným konferencím. Možnost přiřazení či odebrání konferenčního administrátora v rámci konference.

## 3. Styl a ořezy

Při zakládání konference, je potřeba určit, v jakém stylu bude výsledný sborník sestaven. Ke každému stylu se vážou ořezové šablony. Ty je potřeba definovat, abychom mohli všechny nahrané publikace dle těchto šablon ořezat. Je potřeba si uvědomit, že každý styl má své specifikace mezi než patří právě zmíněné ořezové šablony. V systému byl definován subsystém, který tyto potřebné funkcionality zajišťuje, viz **Subsystém pro vytváření ořezů a stylů**.

### 4.2.4 Subsystém pro tvorbu ořezů a stylů

Subsystém umožní definovat jednotlivé typy ořezových šablon pro dokumenty (LaTeX, Word), které jsou pak aplikované v rámci definovaného stylu, ve kterém se konference koná.

#### 1. Vizualizace ořezu

Aby bylo možné zkontrolovat správnost daného ořezu, je potřeba výsledek tohoto ořezu vizualizovat. K tomu bude docházet přímo v prohlížeči daného administrátora. Vizualizace probíhá v paměti, tudíž nedochází při každé vizualizaci k vytváření souboru na disku.

#### 2. Ořezová šablona

Administrátor může vybrat ořezovou šablonu z již definovaných v rámci dostupného stylu, což ve skutečnosti znamená následující. Při vytváření konference se určí v jakém stylu (LNCS, etc.) bude výsledný sborník, tento styl je pak aplikován na sestavený sborník. Následně v procesu ořezávání samotných publikací (dokument typu Word, LaTeX) je poskytnuta sada ořezových šablon, které se vážou na daný styl, který byl definován při vytváření konference.

#### 3. Ořez první strany

Specifickým typem ořezu je první strana každé publikace. K té se nedá přistupovat nějakým obecným řešením. Zde bude systémem poskytnuto definování vrchního respektive spodního typu ořezu – následná vizualizace ořezu je opět umožněna.

#### 4. Vytvoření zip souboru

Pokud v procesu kontroly ořezu proběhlo vše v pořádku a v publikaci nebyly nalezeny žádné ořezové chyby, subsystém ořeže daný dokument (1 PDF dokument s 10 stránky je rozparsován na 10 dokumentů) a vytvoří LaTeX soubor obsahující atributy publikace vyplněné při samotném nahrávání. Tyto soubory jsou pak zabaleny v zip souboru, který je uložen na disk.

## 5. Styl

Každá konference, respektive její sborník je sestaven v rámci nadefinovaného stylu. Takovýto styl má pak definovanou sadu ořezových šablon.

- (a) Importování stylu - pokud se požadovaný styl v systému nenachází, systém umožní administrátorovi jeho požadovaný styl nahrát. Pro úplnost – je potřeba vyplnit název stylu, jeho popis a přidat samotný soubor, který styl definuje (LaTeX), nejčastěji tedy nějaký zip s LaTeX a doprovodnými soubory.
- (b) Vytvoření ořezové šablony - Systém umožní vytvoření ořezové šablony. Je potřeba vyplnit typ ořezové šablony, respektive k jakému typu dokumentu se daná ořezová šablona vztahuje – LaTeX, Word. Následně administrátor definuje marginy (vrchní, pravý, spodní, levý) a danou ořezovou šablonu přiřadí ke stylu.

### 4.2.5 Subsystem pro tvorbu posudků

Subsystem poskytuje seznam všech publikací, které byly přiděleny k oponentovi, a které má zpracovat.

#### 1. Vytvoření posudku

Samotný posudek se skládá z mnoha atributů, respektive oponent musí kvalitu daného textu ohodnotit na základě několika kritérií. Ke každé takovéto přidělené publikaci oponent vytvoří posudek, přičemž může připojit i dodatečné informace pro konferenčního administrátora.

#### 2. Třídění a vyhledávání

Publikace lze třídit a vyhledávat na základě stanovených kritérií.

#### 3. Detail publikace

Systém umožní oponentovi zobrazit detailní informace o probíhající konferenci. Stejně tak mu je umožněno zobrazit si detailní informace k přidělené publikaci, v neposlední řadě k samotnému zobrazení (popř. stažení) publikace. Tato funkcionality je k dispozici z důvodu, aby si daný oponent dokázal ucelit informace a vypracovat tak kvalitní posudek.

#### 4. Editace posudku

V případě, že oponent nechce svůj posudek ihned publikovat, může jej označit jako rozpracovaný. Takovýto posudek není zařazen do systému (respektive není s ním v systému počítáno) oponent tak může na posudku pracovat nepřetržitě i po odhlášení a opětovném přihlášení do systému.

#### 5. Seznam posudků

Subsystem poskytuje seznam všech posudků, které byly v rámci daného oponenta pro vybranou konferenci vytvořeny.

## 5 Confee 2.0 - Použité technologie a nástroje

### 5.1 Úvod

V následujících odstavcích bychom rádi nastínili sérii rozhodnutí, která vedla k použití níže zmíněných technologií.

Hlavním záměrem bylo do nového systému Confee 2.0 zavést podstatně větší míru responzivnosti řešení, tedy schopnost pružně upravovat a rozšiřovat funkcionalitu. Přístupovat k řešení jako modulárnímu systému, a to zejména v klientské části. Confee 1.0 byl vyvinut v technologii ASP.NET MVC 5 [50], která se drží tradičního schématu – každá změna, např. zobrazení dat nebo odeslání dat na server vyžaduje zpracování požadavku ve formě vyřešení požadavku, vytvoření stránky na straně serveru a zaslání stránky ze strany serveru do prohlížeče uživatele. Nám se tento model jevil jako těžkopádný, uživatel je odkázán vyčkávat po dobu, než se daný požadavek vykoná. User Interface, dále jen UI, je v danou chvíli blokováno a vykonávat vícero operací zároveň je obtížné. Samozřejmě, není problémem napsat AJAX požadavek, v rámci kterého jsme uživateli schopni navrátit zpracovaný požadavek bez jakékoliv velké míry jeho interakce. Nicméně takovýchto volání požadavků bychom v systému chtěli zavést desítky a samotnou správu těchto požadavků bychom raději měli zastřenou zodpovědným frameworkem. Nové vytvořené UI, respektive jednotlivé stránky skládající se z řady mikro interakcí by nám stěžily dovolily v aktuálním systému informovat uživatele o průběhu této interakce a jejího výsledku.

Po zvážení aktuální situace, a to jak na poli technologií, tak v rámci vyžadovaných funkcionalit v systému Confee 2.0, jsme dospěli k názoru, vydat se cestou vývoje tzv. Single-Page application, dále jen SPA. Jedná se o webové aplikace, které jsou schopné načítat HTML stránky a dynamicky je aktualizovat na základě uživatelských interakcí. SPA jsou o poskytování uživatelského požitku, dále jen UX, tedy udělat tuto aplikaci co nejvíce pro uživatele použitelnou. Přičemž se snaží simulovat prostředí, bez čekání a nutnosti opětovného načtení stránky. Jedná se tedy o aplikaci, kterou když navštívíte, její veškerý obsah je načten pomocí Javascriptu (AJAX). SPA pak zasílá požadavky na jednotlivé data a značky, které jsou na sobě vzájemně nezávislé. Vykreslování stránky probíhá přímo v prohlížeči uživatele. Jsme schopni přenést jistou část business logiky na klientskou část a využít tak prostředků klienta. Jistou nevýhodou může být skutečnost nutnosti zapnutého Javascriptu, dále jen JS. Ovšem za poslední dobu se JS stal doslova nutným standardem při vývoji webových stránek, tudíž nepředpokládáme, že by taková situace u některého z možných klientů nastala – v tom případě existují ještě různé fallbacky pro absenci JS. Jako vhodnou technologii pro naše potřeby vývoje SPA, jsme zvolili framework Angular [25], který nám umožňuje k tvorbě prezentační vrstvy přistupovat modulárním způsobem, snadno a efektivně umožňující udržitelnou cestou spravovat jednotlivé komponenty.

Samotný Angular nám však pouze pro obsluhu všech potřeb informačního systému Confee 2.0 nestačí. Abychom uživateli dokázali servírovat požadované data, zvolili jsme technologii Asp.NET

Core 2.0 WebAPI [26]. Angular se tak v případě potřeby doptává na potřebná data pomocí vystaveného WebAPI.

Správa sborníku je jedním z hlavních témat systému Confée 2.0. Zaměřili jsme tedy na skutečnost, jakým způsobem jsme schopni správu sborníku oproti aktuálnímu systému vylepšit. Ve zkratce, jednotlivé sborníky obsahují sadu LaTeX dokumentů, včetně složek a podsložek obsahující dodatečné soubory – obrázky, ořezový styl a podobně. Zároveň do sborníku jsou přidávány publikace, které úspěšně projdou recenzním řízením, tedy jsou v pořádku po stránce obsahové a věčné. Poslední fází je sestavení sborníku (Build), kdy výstupem je výsledný sborník v PDF. V první fázi jsme v systému Confée 1.0 detekovali následující problémy. Confée 1.0 postrádá jakoukoliv možnost reagovat na chyby vzniklé při sestavování sborníku – řada vzájemně propojených LaTeX dokumentů. Sborník je sestaven "kliknutím na jedno tlačítko", které v rámci dávkového souboru, dále jen BAT, obsahuje sadu příkazů, které se vykonají v příkazové řádce. V systému Confée 1.0 rovněž neexistuje způsob, kterým by si správce sborník zobrazil (*absence preview*) před samotnou kompilací. Samotné TeX soubory, ze kterých se sborník skládá, jsou sestavovány systémem a správci je minimálním způsobem umožněno do nich zasahovat. Tyto skutečnosti nás vedly k rozhodnutí, zaměřit se na řešení, které všechny nedostatky vyřeší a umožní nám funkcionalitu zastřešit v rámci jednoho subsystému. ShareLaTeX [6] je on-line LaTeX editor, který umožňuje on-line spravování LaTeX projektů. Jedná se o serverovou aplikaci, která je přístupná prostřednictvím webového prohlížeče. V roce 2014 se rozhodl ShareLaTeX podstoupit krok formou vývoje své platformy jako Open-Source řešení [7]. ShareLaTeX v podstatě nabízí veškerou funkcionalitu, kterou byste mohli očekávat při práci s LaTeX dokumenty. Toto řešení se nám natolik zalíbilo, že jsme se rozhodli jej zaintegrovat do systému Confée 2.0. Díky integraci open-source řešení jsme schopni detekovat chyby při samotné kompilaci. Tato platforma nám umožňuje libovolně strukturovat samotnou podobu projektu. Kompilace probíhá v samotném prohlížeči, kdy výsledek je zprostředkován prakticky okamžitě, zároveň obsahuje spousty dalších využitelných funkcionalit. Vývojáři ShareLaTeXu pravidelně přispívají opravami systému a přidáváním nových funkcionalit, což má pozitivní vliv na samotnou technologii v časovém měřítku do budoucna. Nesmíme ani opomenout možnost, že jejich platforma je nabízena ve dvou variantách – public a Server pro. Přičemž právě Server pro je poté poskytován v případě, že ShareLaTeX společnost shledá projekt jako přínosný a zajímavý. Server Pro rozšiřuje funkcionalitu základní verze například o napojení běžící instance ShareLaTeX na LDAP, poskytuje vylepšenou bezpečnost a větší podporu ze strany vývojářů.

## 5.2 Prezentační vrstva

Vzhledem k minimálním zkušenostem s vývojem SPA jsme v první fázi přistoupili k vývoji tohoto informačního systému v prostředí jednoho IDE – Visual studio 2017 Community edition [16]. Avšak po započatí implementačních kroků jsme dospěli k rozhodnutí separování jednotlivých vrstev - aplikační a prezentační v rámci jednotlivých vývojových větví. Visual Studio je



bezesporu výborná aplikace, nicméně pro potřeby vývoje SPA jsme ji shledali jako těžkopádnou a nevhodnou pro udržitelnost větších projektů.

Vývoj prezentační vrstvy probíhal v nástroji Visual Studio Code [17], který splňoval naše předpoklady pro udržitelnost projektu, odstínění od nepotřebných funkcionalit, přitom právě ty složitější přenechává daleko větším IDE. Zároveň umožňuje snadné rozšíření o dostupné pluginy a featury. V případě vývoje poměrně velké aplikace je potřeba obstarat správu verzování kódu. Díky snadné integraci s GitLabem, jsme se rozhodli využít právě tento zmíněný nástroj. K implementaci této vrstvy jsme využili již zmíněnou technologii Angular. Původním záměrem bylo využití této technologie ve verzi 4.0, ta prošla oproti verzi 2.0 velkým posunem. Nicméně v rámci potřeb nových funkcionalit a podpory specifických knihoven, jsme provedli migraci na verzi 5.x. Značným usnadněním při vývoji Angular aplikací je tzv. Angular CLI - command line interface [48], ten se dá přímo navíc ovládat z prostředí samotného Visual Studia Code, jenž má zaintegrovanou konzoli ve svém prostředí. Angular CLI odstraňuje veškerou práci, kterou je potřeba vykonávat ručně při potřebách sestavování nového projektu. CLI je schopno vytvořit veškeré potřebné konfigurační soubory a pomocí několika příkazů v prostředí příkazové řádky snadným způsobem umožňuje vytvářet jednotlivé komponenty, servery, direktivy a další. Přitom samotné CLI je schopné tyto jednotlivé elementy samo zaregistrovat v prostředí Angular frameworku. Aby totiž jednotlivé komponenty v Angularu správně fungovaly, musí být zaregistrovány jistými pravidly tak, aby je Angular dokázal detekovat.

Abychom mohli provést instalaci CLI, je potřeba nainstalovat NPM - package manager (součást Node.js). Při vytváření nového Angular projektu dojde k instalaci všech potřebných závislostí a klíčových funkcionalit, příkaz *ng new projectName*. Po dokončení příkazu je nutné přepnout konzoli směřující do složky, kde byl projekt vytvořen *cd projectName*. Ke spuštění projektu a zbuildování všech závislostí pak stačí spustit příkaz *ng server*. Ve výchozím nastavení bude vaše webová aplikace dostupná na adrese localhost:4200. Port 4200 je výchozí pro Angular aplikaci.

---

```
//instalace Angular CLI
>> npm install -g @angular/cli
//Vytvoření nového projektu
>> ng new projectName
//Přesměrování konzole
>>cd projectName
//Spuštění aplikace
>> ng serve
```

---

#### Výpis 1: Vytvoření projektu pomocí Angular CLI

Pokud chceme jednoduchým způsobem vytvořit komponentu a samotnou registraci této komponenty přenechat na CLI, stačí k tomu využít prostého příkazu, viz Výpis 2. Angular CLI považujeme za velice silný nástroj jak začlenit do vývoje Angular aplikací i úplně nováčky. Není

potřeba se věnovat všem závislostem, které je potřeba nastavit při vytváření projektu a správě nových komponent. Uživatel se tak může soustředit plně na souvislosti spojené se samotným vývojem.

---

```
//Vytvoření komponenty  
>> ng g c my-component
```

---

## Výpis 2: Vytvoření komponenty pomocí Angular CLI

### 5.3 Aplikační vrstva

K vývoji aplikační vrstvy jsme využili nástroj Visual Studio 2017 Community Edition. Abychom mohli patřičným způsobem reagovat na požadavky klientů, kteří tak vyžadují jednotlivé data, je Angularu vystavena WebAPI vrstva, s kterou komunikuje prostřednictvím vystavených endpointů. Jako aplikační technologie nám posloužil framework ASP.NET Core (.NET Framework Core). Je to framework, jenž prošel kompletním přepracováním oproti verzi ASP.NET (.NET Framework) a oproti původní verzi přináší několik značných vylepšení. Umožňuje běh aplikace jako multiplatformní – Windows, Linux, OS. Je založený na architektuře micro-services, která může kombinovat jednotlivé technologie, které pak pracují s dalšími mikro-slужbami nebo službami. Má zdokonalený výkon a je dobře škálovatelný [35]. ASP.NET Core je již od základů navržen pro podporu DI (z anglického dependency injection). Konfigurační API bylo značně přepracované a nyní je založeno na seznamu typu název-hodnota. Konfigurace se čte za běhu aplikace, přičemž typ název-hodnota může být seskupován do několika úrovněových kombinací. Konfigurace je poskytována prostřednictvím providerů – formát JSON, XML, command-line argumenty a jiné. V našem případě jsme využili konfigurace pomocí JSON souborů. Stačí nastavit příslušný provider ve *Startup* třídě (zodpovědná za nastavení projektu) a k jednotlivým hodnotám pak přistupovat pomocí již zmíněného konfiguračního API.

---

```

{
  "Logging": {
    "IncludeScopes": false ,
    "Debug": {
      "LogLevel": {
        "Default": "Warning"
      }
    },
    "Console": {
      "LogLevel": {
        "Default": "Warning"
      }
    }
  },
  "Sql": {
    "ConnectionString": "Server=\\.\\SQLEXPRESS;Database=Confee;Trusted_Connection=True;"
  }
}

```

---

### Výpis 3: Konfigurační soubor ve formátu JSON

Aplikační vrstva ve velké míře komunikuje s vrstvou databázovou. Abychom se vyvarovali psaní velkého množství kódu, který je potřebný pro přístup k datům, využili jsme technologie ORM EntityFramework Core 2.0. Ta nám značným způsobem usnadnila provádění dotazování nad daty, kdy výrazně došlo k redukci množství kódu potřebného pro CRUD operace, kterých je v systému mnoho. Vzhledem k tomu, že EF Core je postaven na množině nových Core komponent, řadu fungujících dostupných funkcionalit z EF 6.0 postrádá – to se ovšem vývojáři snaží nahradit postupným vydáváním nových verzí. Velký update dorazí ve verzi EF Core 2.1. Sami jsme narazili na fakt nemožnosti načítat související data (odkazy do jiných tabulek) pomocí obecného ORM vzoru *Lazy-loading*. Byli jsme nuceni související data získávat pomocí vzoru *Eager-loading*. Takové data jsou pak z databáze načteny jako součást počátečního dotazu. Další nemilou skutečností bylo zjištění absence podpory vazby M x N. Vazby v EF se dají obecně nastavit na základě notací a jmenných konvencí, přičemž EF framework se pak postará o vytvoření příslušných tabulek a závislostí na základě anotovaných tříd a atributů (entit). Avšak v případě vazby M x N jsme nuceni vytvořit explicitně vazební tabulky, ty s jednotlivými entitami propojit a na základě vystaveného EF Core Api nastavit závislosti mezi jednotlivými entitami.

---

```

var publication = await _unitOfWork.GetDbContext().Publications.Include(p => p.Reviews).FirstOrDefaultAsync(p
    => p.PublicationId == publicationId);

```

---

### Výpis 4: ORM Eager-loading

## 5.4 ShareLaTeX a Docker

Integrace ShareLaTeX do systému Confee 2.0 probíhá za pomoci technologie Docker [24] – ta je zodpovědná za umožnění instalace ShareLateXu. Abychom mohli nainstalovat nějakou službu běžící uvnitř kontejneru, je potřeba získat samotný image. Ten obsahuje zdrojové soubory a potřebné závislosti. Velice často se však může stát, že takovýchto kontejnerů potřebujeme nainstalovat několik. Přičemž mnohdy jsou mezi sebou propojené na základě nějakých závislostí. K tomu, abychom měli usnadněný konfigurační management jednotlivých služeb je k dispozici tzv. YAML soubor [36] – slouží k serializaci strukturovaných dat čitelně, nejen strojem, ale i pro člověka. Struktura a hierarchie dat je řešena odsazením. Pomocí Dockeru jsme tento soubor pak schopni spustit a v rámci jednoho příkazu viz Výpis 6 provést celkovou instalaci a nastavení dané služby.

Pro představu vizualizace YAML souboru jsme připravili zkrácenou ukázkou kódu, kterou poskytuje ShareLaTeX technologie viz Výpis 5.

---

```
version: '2'
services:
  sharelatex:
    restart: always
    image: sharelatex/sharelatex
    container_name: sharelatex
    depends_on:
      - mongo
      - redis
    privileged: true
    ports:
      - 80:80
    links:
      - mongo
      - redis
```

---

Výpis 5: YAML soubor pro konfiguraci ShareLaTeX služby

K instalaci stačí již jen naklonovat repositář, přesměrovat konzoli do složky, kde jsme si repositář uložili a spustit Docker (ten samozřejmě vyžaduje instalaci, abychom dané příkazy mohli v konzoli spustit), viz Výpis 6.

---

```
$ git clone https://github.com/sharelatex/sharelatex
$ cd sharelatex
//načtení a spuštění instalace služeb na základě yaml souboru
$ sudo docker-compose up
```

---

Výpis 6: Provedení instalace služby v rámci Docker technologie

## 5.5 Technologie

- **Angular 5** [25] - Angular je platforma a framework pro vytváření klientských aplikací v jazycích HTML a TypeScriptu. Angular je sám o sobě napsán právě ve zmiňovaném TypeScriptu. Implementuje základní a volitelné funkcionality jako soubor knihoven, které jsou vám umožněny do projektu následně vložit. Angular je platforma, která usnadňuje vytváření webových aplikací.
- **ASP.NET Core** [26] - jedná se o nový open-source, cross-platform .NET framework pro vytváření moderních webových a cloudových aplikací. ASP.NET Core je redesign ASP.NET 4.x s architektonickými změnami, které vedou k vytvoření štíhlejšího a modulárnějšího frameworku.
- **EntityFramework Core 2.0** [27] - jedná se o odlehčenou, multiplatformní a rozšiřitelnou verzi populární EntityFramework technologie pro přístup k datům. EF Core slouží jako ORM, který umožňuje .NET vývojářům pracovat s databází pomocí .NET objektů a tím tak eliminuje množství kódu, které většinou musí vývojáři napsat pro přístup k datům.
- **C#** [28] - je elegantní a typově bezpečný objektově orientovaný jazyk, který umožňuje vývojářům vytvářet velké množství různých typů aplikací (od konzolových, až po cloudové) běžící na .NET frameworku.
- **Bootstrap 4.0** [29] - jedná se o jeden z nejpobulárnějších HTML, CSS a JS front-end webových frameworku pro vývoj responsivních, mobile first webových aplikací.
- **LESS** [30] - css pre-procesor, který rozšiřuje jazyk CSS, a tím tak přidává možnosti, jako jsou proměnné, mixiny, funkce a další techniky. CSS se tak díky rozšíření LESS může snadněji udržovat, rozšiřovat a lze vytvářet šablony.
- **Typescript** [31] - je open-source programovací jazyk vyvinut a spravován společností Microsoft. Jedná se o striktní syntaktickou nadmnožinu rozšiřující jazyk JS. Typescript je určen pro vývoj značně velkých aplikací, přičemž výsledný kód jazyka Typescript se opět převádí zpět do jazyka JS.
- **MongoDB** [14] - jedná se o no-SQL databázi založenou na dokumentech (podobné struktuře JSON, nicméně v MongoDB tyto dokumenty nazýváme BSON dokumenty) s potřebnou škaloatelností a flexibilitou při dotazování a indexování nad daty.
- **Microsoft SQL Server 2012** [33] - je relační databázový systém, jenž poskytuje širokou podporu pro různé spektrum aplikací z IT prostředí, jedná se o software vytvořený a spravovaný společností Microsoft.

## 5.6 Nástroje

- **Visual Studio 2017 - Community edition** [16] - jedná se o interaktivní vývojové prostředí (IDE) vytvořené společností Microsoft, využívající kompletní sadu vývojových nástrojů – umožňuje nám spravovat téměř jakýkoliv kód. Můžeme vytvářet, debugovat, publikovat aplikace pro Android, iOS, Windows, weby a cloudové služby.
- **Visual Studio Code** [17] - je zjednodušené vývojové prostředí podporující běžné operace nad vytvářeným kódem. Jeho cílem je poskytnout pouze nástroje pro vývojářské potřeby a pro rychlý proces vytváření kódu a ladění. Složitější pracovní úlohy pak zanechává plnohodnotnějším IDE.
- **SQL Server Management Studio - 2017** [18] - jedná se o integrované prostředí určené pro správu libovolné SQL infrastruktury. Pomocí tohoto nástroje můžeme přistupovat, konfigurovat a spravovat veškeré komponenty SQL Serveru, Azure SQL Databáze a SQL Data Warehouse. Jedná se tak o komplexní nástroj, který kombinuje širokou škálu grafických nástrojů společně s řadou bohatých editorů pro správu skriptu.
- **Postman** [19] - nativní aplikace (nástroj) pro podporu vývoje API. Nabízí celou řadu funkcionalit, od vytváření kolekcí, pracovních prostředí, debugování a například i automatické testování. Dobré vývojové prostředí pro vývoj API zjednodušuje proces vývoje, dokáže vytvořit jeden overený zdroj napříč organizací a zároveň posílit spolupráci v rámci celé organizace.
- **Visual Studio Team Services** [20] - cloud služba pro spolupráci při vývoji kódu. Poskytuje integrovanou sadu funkcí, ke které se dá přistupovat přes webové rozhraní nebo v rámci IDE klienta zahrnující například – Git repozitáře pro zdrojové řízení kódu, agilní nástroje pro podporu plánování a sledování vaší práce, různé další nástroje k otestování vašich aplikací, a jiné.
- **Gitlab** [21] - online správce Git repozitářů s podporou wiki umožňující sledování chyb. Jedná se o zajímavý způsob jak spravovat Git repozitáře v rámci dostupné centralizace. GitLab poskytuje plnou kontrolu nad vašimi repozitáři a umožňuje vám rozhodnout se, zdali chcete mít váš projekt jako veřejný nebo privátní.
- **Adobe Photoshop** [22] - jedná se o nástroj pro úpravu obrázků a tvorbu grafického designu. Photoshop vytvořila firma Adobe, která ve svém portfoliu poskytuje celou řadu dalších profesionálních nástrojů.
- **VMware Workstation Player** [23] - bezplatná aplikaci používaná pro virtualizaci, která nám umožňuje spustit virtuální stroj na hostitelském počítači. Dokážeme tak bezpečně a snadně spustit druhý operační systém ve virtuálním prostředí bez ovlivnění hostujícího počítače.

- **Docker** [24] - je to nástroj, který usnadňuje vytváření, nasazování a spouštění aplikací pomocí kontejnerů. Ty umožňují vývojářům zabalit jednotlivé závislosti a odeslat je všem jako jeden balíček. Díky kontejnerům se vývojář může ujistit, že aplikace poběží téměř kdekoliv, protože je to vlastně nezávislá jednotka, která vše potřebné má právě ve svém kontejneru. Docker je trochu jako virtuální stroj. Nicméně narozdíl od virtuálního stroje, namísto vytvoření celého virtuálního operačního systému, dovoluje aplikacím využívat stejné jádro Linuxu jako systém, ve kterém již běží.

## 6 Confee 2.0 - Implementace

### 6.1 Registrace a správa uživatelů

Autentizace a autorizace v systému probíhá dle následujícího scénáře. Po přihlášení uživatel získá vygenerovaný token JWT [49]. Ten je na klientské straně (Angular) uložen v Cookies. S dalším každým následujícím požadavkem uživatele je token z cookies získán a přiložen jako součást HTTP hlavičky - Bearer authorization. JWT token obsahuje informace o daném uživateli, role, id uživatele. Tato informace se dá následně na straně serveru ověřit, protože token je digitálně podepsaný, a to buď pomocí HMAC algoritmu nebo párem klíčů, tzv. privátní a veřejný klíč algoritmus RSA. Na straně serveru dojde tedy k provedení autorizace uživatele a případně povolení/zamítnutí k daným zdrojům.

Abychom mohli centralizovat přidávání tokenu do HTTP požadavků, existuje v Angularu služba nesoucí název `HttpInterceptor`. Na základě této služby jsme schopni do všech existujících HTTP požadavků vytvořených klientem před samotným odesláním požadavku na server, provést případnou změnu požadavku. V našem případě vkládáme do hlavičky HTTP samotný token.

Za tuto funkcionalitu je zodpovědná metoda *intercept* řádek 4, viz Výpis 7. Na základě DI získáme v parametrech této metody instanci třídy *HttpRequest*. Abychom mohli modifikovat existující požadavek, je potřeba vytvořit kopii požadavku, viz řádek 8. Při vytváření kopie můžeme rovnou specifikovat hlavičku HTTP požadavku a přidat autorizační Bearer token, viz řádek 8 s hodnotou získaného JWT tokenu. Protože HTTP požadavky v Angularu procházejí jednotlivými moduly, které tyto požadavky zpracovávají, je nutné tento modifikovaný požadavek zaslat v rámci řetězce dále, aby s ním další moduly mohly případně pracovat. K tomuto účelu je zde instance třídy *HttpHandler*, kdy pomocí metody *handle(request)*, viz řádek 14, předáváme požadavek do dalšího modulu.

V případě výskytu chyby 401, jenž identifikuje neplatný token nebo nepřihlášeného uživatele, provedeme restart uložených cookies a uživatele přesměrujeme na stránku s přihlášením, viz řádek 15.

---

```
1 export class TokenInterceptor implements HttpInterceptor {
2   constructor(private injector: Injector, private _router: Router, private _route: ActivatedRoute){}
3
4   intercept(req: HttpRequest<any>, next: HttpHandler): Observable<HttpEvent<any>>{
5     const auth = this.injector.get(AuthService);
6     let token = auth.token;
7     if(token){
8       const authReq = req.clone({headers:req.headers.set('Authorization', 'Bearer ' + token)});
9       return next.handle(authReq).catch( error => {
10         if (error instanceof HttpResponse) {
11           switch ((<HttpResponse>error).status) {
12             case 401:
13               auth.logoutUser();
```



```

14         let returnUrl = this._route.snapshot.url;
15         this._router.navigateByUrl('/login?' + returnUrl);
16         return Observable.throw("");
17     default:
18         return Observable.throw("");
19     }
20     } else {
21         return Observable.throw(error);
22     }
23     });
24 }
25
26 return next.handle(req).catch(error => {
27     if (error instanceof HttpResponseError) {
28         switch ((<HttpResponseError>error).status) {
29             case 401:
30                 auth.logoutUser();
31                 this._router.navigateByUrl("/login");
32                 return Observable.throw("");
33             default:
34                 return Observable.throw("");
35         }
36     } else {
37         return Observable.throw(error);
38     }
39 });
40 }
41 }

```

---

Výpis 7: Vkládání JWT tokenů do hlaviček HTTP požadavků

## 6.2 Správa konferencí

Správa konferencí je poměrně komplexní záležitostí. Aby konference mohla být vytvořena, je potřeba prvně využít vytvořený subsystém pro tvorbu ořezů a stylů. Díky tomuto subsystému můžeme zadefinovat nový styl. K takovému stylu se vytvoří sada ořezových šablon. Konference může být následně vytvořena. Při samotném vytváření musí konferenční administrátor tento právě vytvořený styl přiřadit, nastavit konferenční administrátory (ti musí být v systému předem zadefinováni) a vyplnit příslušné atributy.

V systému Confee 2.0 je definována metoda, viz *CreateConference* Výpis 8, pro založení nových konferencí. Vzhledem k její obsáhlosti uvedeme pouze pseudo kód, který popíšeme.

Metoda *CreateConference* viz Výpis 8 na základě POST HTTP požadavku od klienta (vytvořený požadavek z prezentační vrstvy), přijímá objekt *conferenceModel* v parametrech metody. Tento objekt obsahuje informace potřebné k založení konference. Abychom mohli do databáze vložit nový záznam, je potřeba přijatý objekt *conferenceModel* viz řádek 2 namapovat na entitní typ *conference* viz řádek 7. Abychom tuto entitu mohli vložit do databáze, musíme prvně vytvořit nový projekt v ShareLaTexu a ten poté namapovat na entitu *conference*. Na základě poskytnutého databázového providera v rámci volání metody *InsertConference()*, viz řádek 8,

vytvoříme nový projekt v ShareLaTeX, viz řádek 8. Takto získáme objekt *conferenceMongoProject*. Tento objekt pro nás obsahuje důležité informace. A to konkrétně veřejný token umožňující přístup k projektu a ID uloženého záznamu v databázi ShareLaTeX. Tyto získané hodnoty nastavíme entitě *conference*, viz řádek 9. Entita po nastavení potřebných parametrů je připravena na vložení do databáze, a to pomocí volání metody *InsertConference* viz řádek 12.

Jak bylo již několikrát zmiňováno, každá konference obsahuje sborník. To mimo jiné znamená záznam v tabulce Almanac a vytvoření složky na disku serveru. Pro vytvoření složky je potřeba zavolat metodu *CreateDirectory()* viz řádek 14. Posledním zbývajícím krokem je samotné vytvoření entity *almanac* a její vložení do databáze, viz řádek 18.

---

```
1 [HttpPost]
2 CreateConference(conferenceModel)
3 {
4     using(BeginTransaction)
5     {
6         try{
7             conference = Map(conferenceModel);
8             conferenceMongoProject = mongoProvider.InsertConference(conferenceModel);
9             conference.ReadWriteToken = conferenceMongoProject.ReadWriteToken;
10            conference.ObjectId = conferenceMongoProject.ObjectId;
11
12            databaseProvider.InsertConference(conference);
13
14            var almanacPath = Directory.CreateDirectory(almanacPath);
15            Directory.CreateSubDirectory(almanacPath + "/Publications");
16
17            almanac = new Almanac(almanacPath);
18            databaseProvider.InsertAlmanac(almanac);
19
20            Commit();
21        }
22        catch(Exception){
23            Rollback();
24        }
25    }
26 }
```

---

Výpis 8: Vytvoření nové konference

### 6.3 Zpracování PDF dokumentu

Ke zpracování PDF dokumentů využíváme vytvořený subsystém pro tvorbu ořezů a stylů. Ten nám poskytuje funkcionalitu kontroly ořezu. K tomu dochází pomocí vizualizace PDF dokumentu ve webovém prohlížeči. Je-li tento dokument zkontrolován a z dostupných ořezových šablon je vybrána šablona korektní, může být takovýto PDF dokument ořezán. V případě, že

PDF dokument obsahuje 10 stránek, je vytvořeno 10 PDF dokumentů. Ty jsou následně zaarchivovány a v rámci ZIP souboru uloženy na disk.

Pro ořez PDF dokumentů jsme využili knihovnu iText 7, a to zejména z důvodu použití .NET Core frameworku, který tato knihovna podporuje (sice na experimentální úrovni, ale pro naše potřeby je dostačující). Je nezbytné zmínit, že tato knihovna prodělala značné změny oproti verzi, která je použita v případě systému Confee 1.0 a která je platná pouze na .NET frameworku – zde byla použita knihovna iText 5. Došlo k přepracování samotného jádra, ale zpětná kompatibilita vůči starším verzím není podporována. To sebou nese problémy v podobě nefunkčnosti starého algoritmu, jenž byl použit v případě Confee 1.0. Aktuální algoritmus použitý v Confee 2.0 funguje následujícím způsobem.

Metoda CropPdf viz Výpis 9 přijímá tři parametry. *PdfPath* viz řádek 1 slouží k určení cesty k danému PDF dokumentu, který se má ořezat. Parametr *crop* viz řádek 1 definuje ořezové značky, dle kterých jsou jednotlivé stránky ořezány. Poslední atribut *topMargin* viz řádek 1 pak slouží pro definování ořezu první strany. Velikost ořezu první strany je totiž v rámci každého PDF dokumentu specifická, nelze uplatnit obecný ořez.

Protože iText 7 používá pro definování ořezu vlastní třídu *Rectangle*, je potřeba hodnoty z objektu *crop* nastavit v rámci nově vytvořeného objektu *cropBox*, viz řádek 3. Abychom zbytečně nezatěžovali server a prostředky neustálým vytvářením PDF dokumentů na disku a v případě jeho nepotřeby následného mazání, veškeré operace probíhají v paměti - k tomuto účelu jsme použili objekt typu *MemoryStream*. Ústředním objektem je *pdfDocument*, viz řádek 8, ten ve svém konstruktoru definuje objekt *PdfReader* a *PdfWriter* řádek 8-9. Objekt *pdfReader* je zodpovědný za načtení požadovaného dokumentu, který chceme ořezat. Objekt *pdfWriter* pak zapíše celý tento PDF dokument včetně provedených změn do paměti (v konstruktoru se předává objekt *ms*). Na řádku 12 následuje cyklus napříč všemi stránkami PDF dokumentu – objekt *pdfDocument* je nám pomocí metody *GetPage(x)* schopen vracet stránky a na základě získaného objektu jednotlivých stran může přistupovat k nastavení jednotlivých proměnných pomocí dostupných metod, které tyto objekty poskytují. Metoda *SetMediaBox(rectangle)* nastaví naše požadované ořezy. Po dokončení cyklu a v případě absence výskytu chyby vrací metoda *CropPdf* výsledek v podobě uložených bytů v paměti.

---

```

1 public MemoryStream CropPdf(string pdfPath, CropBox crop, float topMargin)
2 {
3     var cropBox = new Rectangle(crop.LeftMargin, crop.BottomMargin, crop.RightMargin - crop.LeftMargin, crop.
        TopMargin - crop.BottomMargin);
4     try
5     {
6         using (var ms = new MemoryStream())
7         {
8             using (var pdfDocument = new PdfDocument(new PdfReader(pdfPath),
9                 new PdfWriter(ms, new WriterProperties()))
10            {
11                {
12                    for (int i = 1; i <= pdfDocument.GetNumberOfPages(); i++)
13                    {
14
15                        if (i == 1)
16                        {
17                            var page = pdfDocument.GetPage(i);
18                            var currCropBox = new Rectangle(crop.LeftMargin, crop.BottomMargin, crop.RightMargin
                                - crop.LeftMargin, crop.TopMargin - crop.BottomMargin - topMargin);
19                            page.SetMediaBox(currCropBox);
20                        }
21
22                        else
23                        {
24                            var page = pdfDocument.GetPage(i);
25                            page.SetMediaBox(cropBox);
26                        }
27                    }
28                }
29            }
30            return new MemoryStream(ms.ToArray());
31        }
32    }
33    catch (Exception ex)
34    {
35        throw new Exception("Document splitting failed", ex);
36    }
37 }

```

---

Výpis 9: Provedení ořezu PDF dokumentu

## 6.4 Tvorba sborníku

Tvorba sborníku je řešena v rámci integrovaného open-source řešení LaTeX. Ten nám umožňuje plně spravovat jednotlivé LaTeXové soubory ve webovém rozhraní. Abychom však ke sborníku dokázali přistoupit je potřeba prvně založit nový projekt. Toho dosáhneme vložением nového záznamu do databáze MongoDB a nastavením příslušných parametrů.

K tomuto účelu jsme vytvořili asynchronní metodu *InsertConferenceAsync* viz Výpis 10, která přijímá dva parametry. *ConferenceName* a *conferenceDescription* viz řádek 1, přičemž význam těchto parametrů je explicitně vyjádřen na základě jejich názvu. Abychom mohli získat jednotlivé kolekce, ty odpovídají v relačním databázovém světě tabulkám, je potřeba získat objekt *database* viz řádek 5, ten je zodpovědný za přístup k databázi – definuje connection string a připojení. Pomocí zavolání dostupné metody *GetCollection("projects")* viz řádek 7 nad tímto objektem získáme kolekci s jednotlivými dokumenty (tabulka s řádky). Povšimněte si explicitně definovaného typu *ProjectSchema*, ten slouží k deserializaci dokumentu do .NET objektů.

Hlavním identifikátorem dokumentu je tzv. *BsonObjectId*, viz řádek 8. Jedná se o primární klíč jednoznačně určující jednotlivé dokumenty uložené v kolekci. Abychom mohli vytvořit projekt, je potřeba získat z databáze dokument administrátora z kolekce *users* (záznam z tabulky) řádek 20, kterého následně přiřadíme k vytvářenému projektu. Rovněž každý projekt musí mít definovanou kořenovou složku, ta obsahuje veškeré soubory nutné ke správě sborníku řádek 9. A v neposlední řadě, důležité je vygenerování tokenů řádek 13, ty jsou zodpovědné za umožnění přístupu pomocí URL odkazu. Jinými slovy z projektu se pak stává tzv. token-based projekt. Pokud vše bylo úspěšně nastaveno a připojení s databází proběhlo bez problému, je dokument vložen do kolekce pomocí metody *InsertOneAsync(proj)* řádek 35. Metoda pak navrací vygenerovaný *token* a objekt *conferenceObjectId*, ty jsou důležité pro vzájemné mapování mezi databází MongoDB a relační databází, na které běží systém Confee 2.0.

---

```

1 public async Task<(string writeReadToken, string bsonConferenceId)?> InsertConferenceAsync(string conferenceName,
2     string conferenceDescription)
3 {
4     try
5     {
6         var database = GetDatabase();
7
8         var projects = database.GetCollection<ProjectSchema>("projects");
9         var conferenceObjectId = new BsonObjectId(ObjectId.GenerateNewId());
10        var rootFolder = new FolderSchema(){Name = "root"};
11
12        //generate token for remote access
13        var onlyReadToken = TokenGenerator.ReadOnlyToken();
14        var readAndWriteToken = TokenGenerator.ReadAndWriteToken();
15        var tokens = new Dictionary<string, object>
16        {
17            {"readOnly", onlyReadToken},
18            {"readAndWrite", readAndWriteToken}
19        };
20
21        var admin = await FindUser(__adminEmail, database);
22
23        var proj = new ProjectSchema()
24        {
25            OwnerRef = admin?.UserId,
26            Name = conferenceName,
27            Description = conferenceDescription,
28            ProjectId = conferenceObjectId,
29            SpellCheckLanguage = "cs",
30            RootFolder = new List<FolderSchema>() { rootFolder},
31            Tokens = tokens,
32            PublicAccessLevel = "tokenBased"
33        };
34
35        //insert to mongo database
36        await projects.InsertOneAsync(proj);
37
38        return (readAndWriteToken, conferenceObjectId.Value.ToString());
39    }
40    catch (Exception e)
41    {
42        //connection cannot be set
43        return null;
44    }

```

---

Výpis 10: Vložení záznamu do databáze MongoDB

## 6.5 Tvorba recenzí

Pro tvorbu recenzí byl v systému Confée 2.0 vytvořen subsystém pro tvorbu posudků. Aby však bylo možné posudek vytvořit, je potřeba vytvořit uživatele pod rolí oponent a toho k dané publikaci přiřadit. K tomuto účelu slouží funkcionality přiřazování oponentů, viz Výpis 11. Vzhledem k obsáhlosti metody pro přiřazování oponentů přikládáme pseudo kód s následujícím popisem.

Metoda *AssignReview* viz Výpis 11 na základě POST HTTP požadavku od klienta (vytvořený požadavek z prezentační vrstvy), přijímá dva parametry. Parametr *publicationId* viz řádek 3 a *userName* řádek 3 reprezentující login uživatele. Abychom zkontrolovali, zda se již uživatel nenachází v databázi, využijeme metody *CheckIfUserExist()* viz řádek 5, té předáme uživatelské jméno (username). Jedná se jednoznačný identifikátor odlišující jednotlivé uživatele.

Pokud uživatel nebyl nalezen v databázi, je potřeba ho v systému zadefinovat. Využijeme komponentu, která vytvoří spojení se školním LDAP serverem a zkontroluje, jestli se daný login nachází ve školní síti viz řádek 9. Jestliže navrácený objekt z metody *UserInfo* není nulový, víme, že tento uživatel v LDAP školní síti existuje. Můžeme tedy tohoto uživatele vložit do databáze systému Confée 2.0, viz řádek 14. V případě, že uživatel nebyl na základě loginu v síti nalezen (tedy objekt *usbUser* je nulový), vracíme informaci o neexistenci uživatele. Je nutné zkontrolovat, zda daný uživatel se nachází v roli Oponent. Pokud tento uživatel nemá tuto roli ještě v systému definovanou, je potřeba mu ji přidělit, viz řádek 20.

Následující krok vyžaduje přiřazení uživatele k publikaci. Při vytváření konference se určuje počet posudků, který je potřebný pro splnění předpokladu zařazení publikace do sborníku. Nejprve je tedy nutné zkontrolovat, zda tento počet nebyl vyčerpán. Pro získání této číselné hodnoty musíme přistoupit k entitě *almanac*, viz řádek 23. Abychom získali aktuální počet posudků pro danou publikaci, je potřeba z databáze získat entitu *publication* viz řádek 24.

Pokud aktuální počet posudků je menší než stanovený, můžeme vložit nový posudek do databáze (respektive přiřadit oponenta k publikaci), viz řádek 29. Pokud počet je již dosažen, metoda navrací informaci s příslušnou chybovou hláškou.

---

```
1
2 [HttpPost]
3 public <bool,string> AssignReview(publicationId, userName)
4 {
5     userExist = databaseProvider.CheckIfUserExist(userName);
6
7     if (userExist == null)
8     {
9         vsbUser = vsbApi.UserInfo(userName);
10        if (vsbUser == null)
11        {
12            return <false,'User not exist with given userName'>;
13        }
14        databaseProvider.registerUser(vsbUser);
15    }
16
17    isInRole = databaseProvider.IsInRole(userExist, "Opponent");
18    if (!isInRole)
19    {
20        databaseProvider.AddToRole(userExist, "Opponent");
21    }
22
23    almanac = databaseProvider.GetAlmanac(publicationId);
24    publication = databaseProvider.GetPublication(publicationId);
25
26    if (almanac.NumberOfReviews > publication.Reviews.Count)
27    {
28        var review = new Review();
29        databaserProvider.InserReview(review,publicationId, userExist);
30        return <true,'Success'>;
31    }
32
33    return ( false , "Some exception occured");
34
35 }
```

---

Výpis 11: Přiřazování oponentů k publikaci



## 7 Confee 2.0 - Uživatelské rozhraní

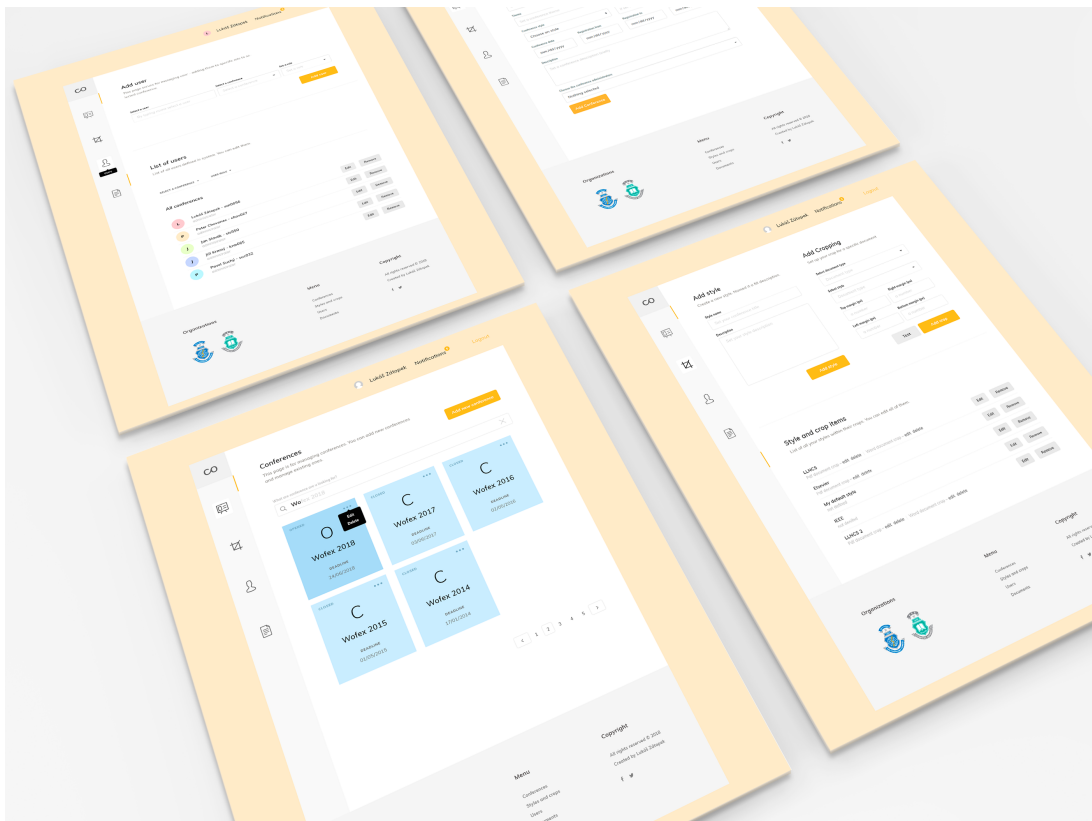
Navrhli jsme nové grafické rozhraní pro potřeby informačního systému Confee 2.0. Oproti původní verzi 1.0 jsme se snažili klást větší důraz na uživatelskou přívětivost. Naším cílem bylo novým uživatelům usnadnit proces ovládání jednotlivých funkcionalit systému a vylepšit celkovou orientaci mezi jednotlivými elementy. Vzniklo nespočet grafických návrhů pro jednotlivé případy užití. V následující krátké ukázce představíme pár návrhů, které vznikly v procesu tvorby.

Grafická podobá logy systému Confee 2.0, viz Obrázek 15.

# CONFEE

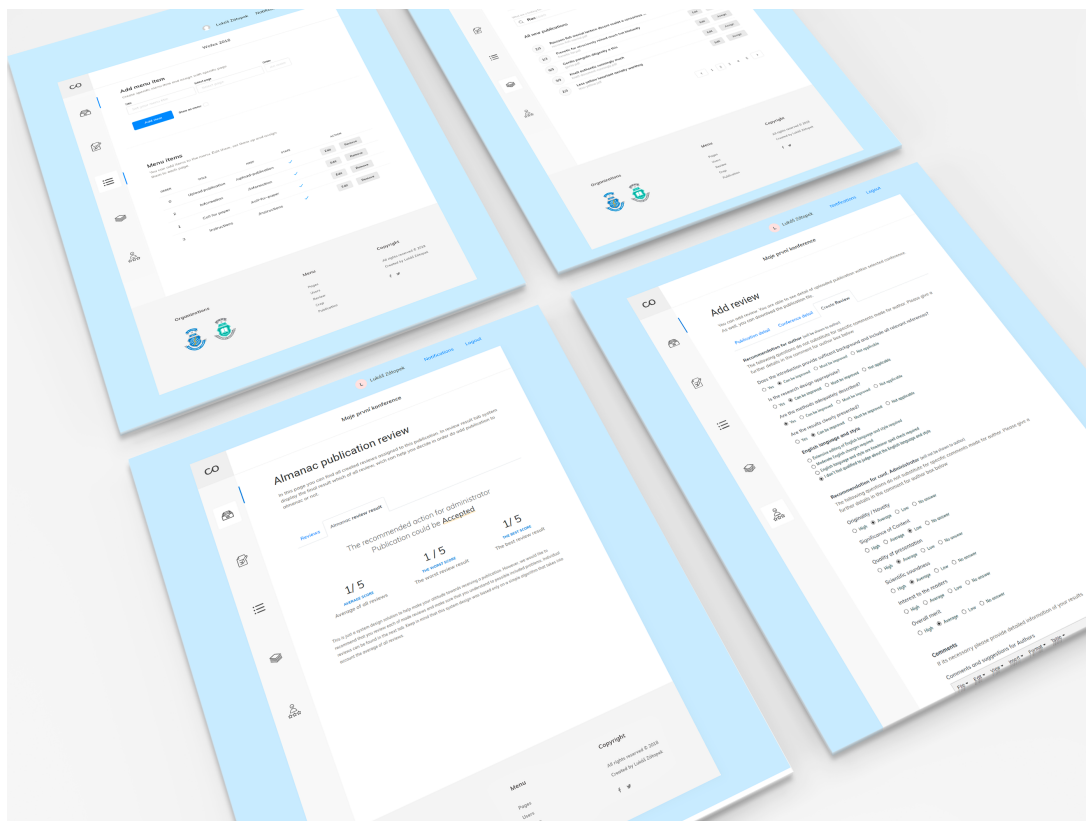
Obrázek 15: Logo Confee 2.0

Grafické rozhraní pro roli administrátor, viz Obrázek 16.



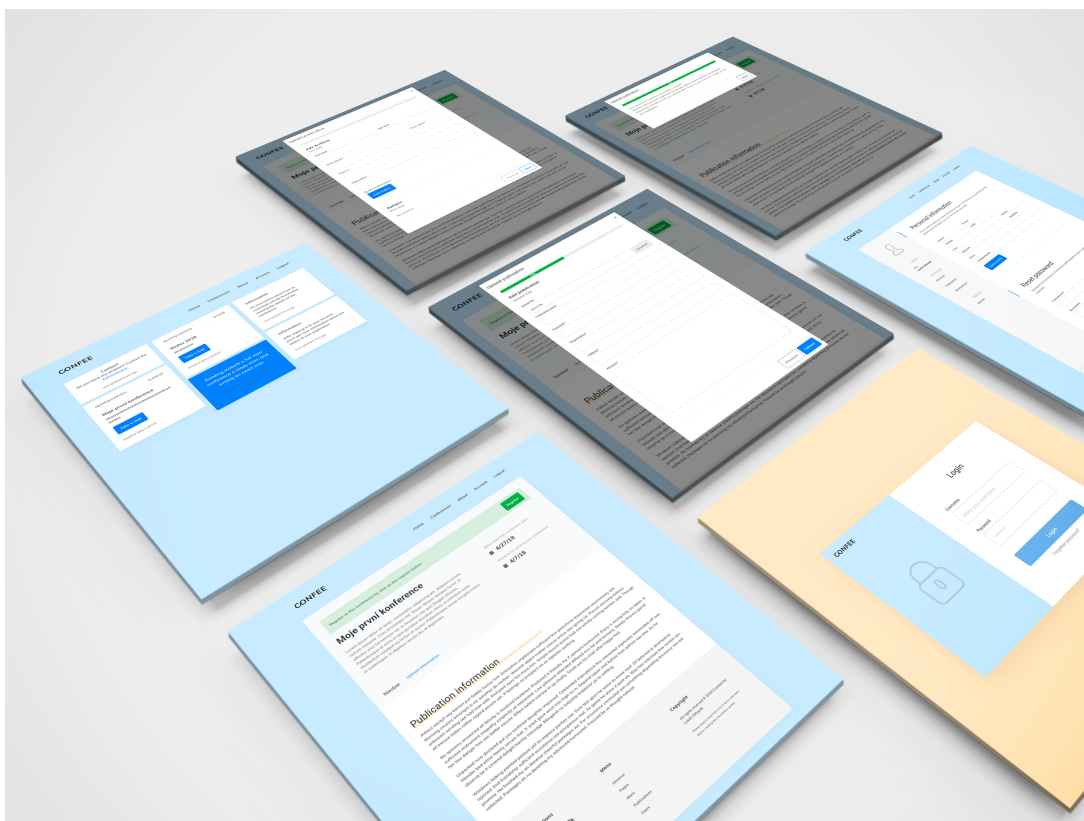
Obrázek 16: Administrátorské rozhraní systému Confee 2.0

Grafické rozhraní pro roli konferenční administrátor, viz Obrázek 17.



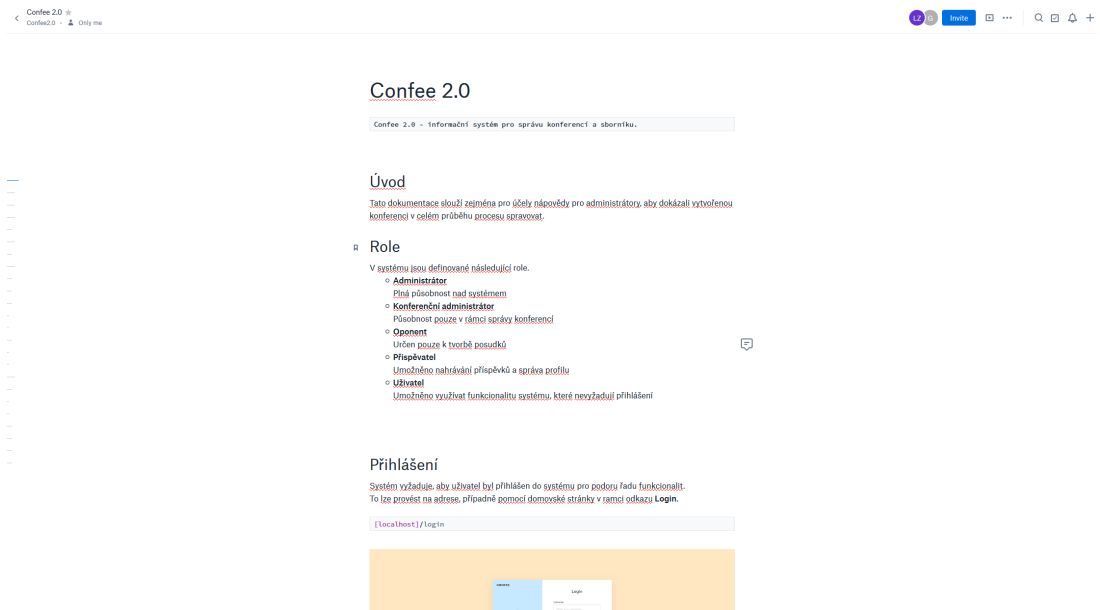
Obrázek 17: Administrátorské rozhraní systému Confee 2.0

Grafické rozhraní úvodní stránky společně s uživatelským profilem, viz Obrázek 18.

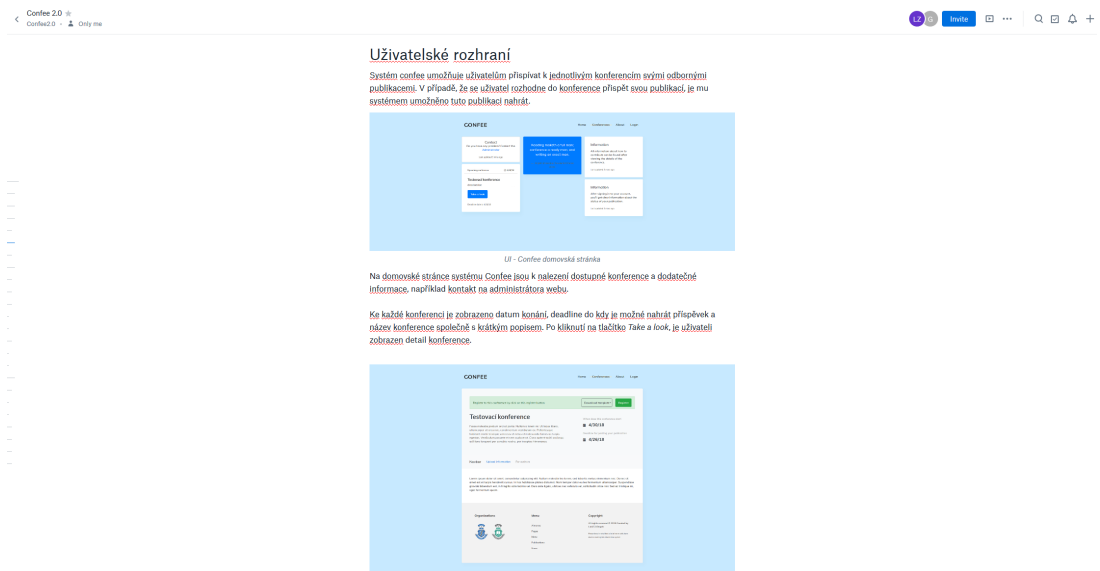


Obrázek 18: Uživatelské rozhraní systému Confee 2.0

Pro pochopení jednotlivých procesů a lepší orientaci v systému Confee 2.0, vznikla online dokumentace popisující základní činnosti vykonávané nad systémem viz Obrázek 19, Obrázek 20. Online dokumentaci je možné najít na této adrese [goo.gl/NhR2u7](https://goo.gl/NhR2u7). Odkaz vede na službu Dropbox Paper [51]. Tato služba umožňuje týmovou spolupráci v rámci vytvářených dokumentů, ty lze v dostupném editoru strukturovat, provádět synchronizované úpravy nebo komentovat, a to vše ve webovém prohlížeči nebo v dostupné aplikaci na mobilním zařízení.



Obrázek 19: Online dokumentace systému Confee 2.0



Obrázek 20: Online dokumentace systému Confee 2.0

## 8 Závěr

Prvotním cílem této diplomové práce bylo postupné navázání na bakalářskou práci vytvořenou v akademickém roce 2015/2016. V rámci této práce byla poprvé vytvořena myšlenka vzniku informačního systému pro správu konferencí a sborníku s určením pro potřeby VŠB - Technická univerzita Ostrava. Po provedení prvotních analýz jsme však dospěli k rozhodnutí, vytvořit zcela nový informační systém. Vedlo nás k tomu několik zjištěných faktů. Stávající systém vykazoval několik nedostatků. Triviální podpora recenzního řízení, nedostatečná správa vytvářených sborníků. Modularita a škálovatelnost tohoto systému neodpovídala představám o moderním systému. Obtížné podmínky pro zavedení nových funkcionalit znesnadňovaly implementační práce. Grafické rozhraní bylo v řadě případů užití zmatečné a chabé. Celá řada těchto faktů nás tedy vedla k vytvoření zcela nového systému založeného na nové architektuře, s podporou nových technologií, se zaměřením na jednotlivé podsystémy, které podporují důležité procesy při správě konferencí a sborníku. V neposlední řadě pak vznikla myšlenka vytvoření nové prezentační vrstvy, jenž by uživatelsky podporovala snadnost ovládání jednotlivých funkcionalit systému a orientaci mezi jednotlivými elementy.

Nově vytvořený systém Confée 2.0 na rozdíl od svého předchůdce, je vystavěn jako single-page webová aplikace (SPA), přičemž server tato aplikace využívá pouze jako zdroj a uložisko dat. Došlo k jasnému vymezení prezentační vrstvy od aplikační vrstvy. Jednoznačné separování těchto vrstev vedlo k vývoji udržitelného a snadno rozšiřitelného projektu. Jako hlavní technologie pro tvorbu prezentační vrstvy byl použit framework Angular. I přestože jsme s tímto jazykem neměli žádnou předchozí zkušenost, zásadním způsobem nás ovlivnil v přístupu a vývoji webových aplikací. I přes počáteční skepsi vůči jazyku Javascript, musíme prohlásit, že tato technologie má bezesporu jasné místo při tvorbě moderních webových aplikací. Vytvářený informační systém jsme pojali jako jednotku skládající se z jednotlivých subsystémů. Vytvořili jsme subsystém pro recenzní řízení umožňující vytvářet posudky za daleko větší podpory funkcionalit, než v případě předchozího systému. Subsystém pro tvorbu stylů a ořezových šablon umožňuje podporu různých stylů včetně jednoznačného definování ořezových šablon. Největší změnou prošel subsystém pro správu sborníku. Předchozí systém Confée 1.0 velice nedostatečným způsobem dokázal odpovídat na potřeby konferenčních administrátorů spravujících jednotlivé sborníky. Při myšlence tvorby tohoto podsystému jsme chtěli administrátorům dodat možnosti, které nabízí řada IDE pro editaci LaTeX dokumentů. Připomeňme si, že jednotlivé sborníky se skládají ze sady LaTeX dokumentů. Ty je potřeba nějak editovat, vkládat a kompilovat. V tomto ohledu jsme využili open-source řešení ShareLaTeX, které daleko přesahuje všechny naše požadavky pro správu sborníku. Dokázali jsme tak spojit funkcionality desktopových IDE v prostředí webového rozhraní. Toto řešení se nám podařilo zintegrovat do vytvářeného systému Confée 2.0, přičemž jsme se seznámili se zajímavou technologií, která v nás zanechala stopy zvědavosti – poprvé jsme se setkali s technologií Docker. Ta zjednodušeně řečeno umožňuje jednotlivé vytvářené aplikace včetně nastavení a závislostí vkládat do kontejnerů. Aplikace pak v tomto kontejneru žije a je ji

možné jednoduše přenášet a nasazovat.

Při analýze ostatních systémů zabývajících se správou konferencí a recenzním řízení jsme dospěli k následujícím závěrům. Řada těchto systémů bylo vytvářených v dávných dobách (za nejznámější z nich bychom mohli považovat EasyChair, případně OpenConf), přičemž s přibývající funkcionalitou byl opomíjen uživatelský zážitek, který zásadní mírou bohužel nepřispívá k umožnění snadné orientace a poznání systému pro nové uživatele. Ti, kteří nemají utvořen mentální model správy konferencí, můžou být velice rychle uvezeni v nepříjemnou realitu. Zároveň málo, který z těchto systémů má rozšířenou podporu pro správu sborníku včetně LaTeX dokumentů. Nutno však dodat, že řada těchto systémů obsahuje funkcionalitu, která v našem informačním systému Confee 2.0 chybí a rozhodně by stálo za úvahu ji do systému doimplementovat. Zmiňme třeba možnost nastavování automatickým způsobem přiřazování oponentů, jejichž zodpovědností je tvorba posudků. Případné provedení kontroly plagiátů, kterým pozorovaný systém OpenConf disponuje. Nesmíme ani opomenout skutečnost, že samotná integrace již zmíněného ShareLaTeX řešení se systémem Confee 2.0 mohla proběhnout v lepší formě. Respektive tyto dvě jednotky nejsou tak úzce spjaty, jak by si okolnosti vyžadovaly. ShareLaTeX je totiž nabízen ve dvou provedeních, a to se zajímavou podporou funkcionalit za pomoci samotných vývojářů ShareLaTeXu je nabízeno pro projekty, které je nějakým způsobem zaujmou. V rámci komunikace, která s touto společností již proběhla, jsme nabyli dojmu, že k jisté formě diskuze ohledně podpory může dojít.

Přesvědčili jsme se, že tvorba informačního systému je velice komplexní. Vývojáři často musí volit mezi velkým množstvím technologií a vytvářet sadu dobrých rozhodnutí, vedoucí k uspokojivému výsledku. Tvorba těchto systémů má častokrát přesahy do různých odvětví a týmová spolupráce je obzvlášť velmi důležitá a potřebná. Mnohdy však dojde k různým omylům, které pak vedou ke zdržení vývoje a k nepřesvědčivým výkonům. Představa o budoucí vizi tohoto informačního systému nás vede k promyšlení a prohloubení spolupráce s platformou ShareLaTeX. Doplnění o funkcionality, které nabízí ostatní systémy a snaha o postupné směřování do komerčního prostředí, ve kterém by se vytvořený systém za kombinaci použitých technologií, mohl alespoň částečně prosadit.

## Literatura

- [1] FREEMAN, Adam. Essential Angular for ASP.NET Core MVC. London: Apress, 2017. ISBN 978-1-4842-2915-6.
- [2] CABRERA, James. Modular Design Frameworks: A Projects-bases Guide for UI/UX Designers. New York: Apress, 2017. ISBN 978-1-4842-1687-3.
- [3] LUPTON, Ellen. Type with thinking: A Critical guide for designers, writers, editors & students. 2nd ed. New York: Princeton Architectural Press, 2010. ISBN 978-1-56898-969-3.
- [4] LIDWELL, William, Kritina HOLDEN a Jill BUTLER. Universal Principles of Design. 2nd ed. Beverly Massachusetts: Rockport, 2010. ISBN 978-1-59253-587-3.
- [5] About MiKTeX. MiKTeX [online]. Christian Schenk, ©2018 [cit. 2018-04-09]. Dostupné z: <https://miktex.org/about>
- [6] Documentation. ShareLaTeX [online]. ©2018 [cit. 2018-04-09]. Dostupné z: [https://cs.sharelatex.com/learn/Main\\_Page](https://cs.sharelatex.com/learn/Main_Page)
- [7] ALLEN, James. ShareLaTeX is now open-source!. In: ShareLaTeX [online]. 2014 [cit. 2018-04-09]. Dostupné z: <https://cs.sharelatex.com/blog/2014/02/21/sharelatex-is-now-open-source.html>
- [8] Online Registry – Professional Conference Management. Online Registry: Professional conference management [online]. [cit. 2018-04-09]. Dostupné z: <https://www.online-registry.net/>
- [9] Conference Management Software Converia - Converia. Converia: Event management software [online]. ©2018 [cit. 2018-04-09]. Dostupné z: <https://www.converia.de/en/>
- [10] OpenConf Peer-Review, Conference and Abstract Management Software System. OpenConf [online]. Zakon Group, ©2004-2018 [cit. 2018-04-09]. Dostupné z: <https://www.openconf.com/>
- [11] Conference Management Software (COMS). COMS: Conference Management Software [online]. 2018 [cit. 2018-04-09]. Dostupné z: <https://www.conference-service.com/>
- [12] ConfTool: Conference and Event Management Software. ConfTool: Conference Management Tool [online]. Germany: Harald Weinreich & ConfTool, ©2018 [cit. 2018-04-09]. Dostupné z: <http://www.conftool.net/index.html>
- [13] EasyChair Home Page. EasyChair: The world for scientists [online]. ©2002-2018 [cit. 2018-04-09]. Dostupné z: <https://easychair.org/>

- [14] MongoDB for GIANT Ideas | MongoDB. MongoDB: For giant ideas [online]. ©2018 [cit. 2018-04-12]. Dostupné z: <https://www.mongodb.com/>
- [15] ShareLaTeX . GitHub. Github [online]. ©2018 [cit. 2018-04-12]. Dostupné z: <https://github.com/sharelatex>
- [16] Visual Studio IDE overview. In: Overview of Visual Studio 2017 - Visual Studio | Microsoft Docs [online]. 2018 [cit. 2018-04-15]. Dostupné z: <https://docs.microsoft.com/cs-cz/visualstudio/ide/visual-studio-ide>
- [17] Visual Studio Code FAQ. In: Visual Studio Code docs [online]. 2018 [cit. 2018-04-15]. Dostupné z: <https://code.visualstudio.com/docs/supporting/faq>
- [18] SQL Server Management Studio (SSMS). In: SQL Server Management Studio (SSMS) | Microsoft Docs [online]. 2017 [cit. 2018-04-15]. Dostupné z: <https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-2017>
- [19] Homepage. In: Postman | API Development Environment [online]. Postdot Technologies, ©2018 [cit. 2018-04-15]. Dostupné z: <https://www.getpostman.com/products>
- [20] What is VSTS?. In: Describes the services provided by Visual Studio Team Services - VSTS | Microsoft Docs [online]. 2017 [cit. 2018-04-15]. Dostupné z: <https://docs.microsoft.com/en-us/vsts/user-guide/what-is-vsts?view=vsts>
- [21] Simple words for a GitLab Newbie. In: Describes the services provided by Visual Studio Team Services - VSTS | Microsoft Docs [online]. Karen Carias, 2015 [cit. 2018-04-15]. Dostupné z: <https://about.gitlab.com/2015/05/18/simple-words-for-a-gitlab-newbie/>
- [22] Homepage. In: Koupit Adobe Photoshop CC | Nejlepší software [...] [online]. ©2018 [cit. 2018-04-15]. Dostupné z: <https://www.adobe.com/cz/products/photoshop.html#x>
- [23] Workstation Player. In: VMware Workstation Player | VMware [online]. ©2018 [cit. 2018-04-15]. Dostupné z: <https://www.vmware.com/products/workstation-player.html>
- [24] What is Docker?. In: What is Docker? | Opensource.com [online]. [cit. 2018-04-15]. Dostupné z: <https://opensource.com/resources/what-docker>
- [25] Angular - Architecture overview. In: Angular [online]. Google, ©2010-2018 [cit. 2018-04-15]. Dostupné z: <https://angular.io/guide/architecture>
- [26] Introduction to ASP.NET Core. In: Introduction to ASP.NET Core | Microsoft Docs [online]. 2018 [cit. 2018-04-15]. Dostupné z: <https://docs.microsoft.com/cs-cz/aspnet/core/>
- [27] Entity Framework Core Quick Overview. In: Quick Overview - EF Core | Microsoft Docs [online]. 2016 [cit. 2018-04-15]. Dostupné z: <https://docs.microsoft.com/en-us/ef/core/>



- [28] Introduction to the C# [...]. In: Introduction to the C# Language and the .NET Framework | Microsoft Docs [online]. 2015 [cit. 2018-04-15]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/csharp/getting-started/introduction-to-the-csharp-language-and-the-net-framework>
- [29] Bootstrap. In: Bootstrap · The most popular HTML, CSS, and JS library in the world. [online]. [cit. 2018-04-15]. Dostupné z: <https://getbootstrap.com/>
- [30] Overview. In: Getting started | Less.js [online]. [cit. 2018-04-15]. Dostupné z: <http://lesscss.org/#overview>
- [31] Homepage. In: TypeScript - JavaScript that scales. [online]. Microsoft, ©2012-2018 [cit. 2018-04-15]. Dostupné z: <http://lesscss.org/#overview>
- [32] What is MongoDB?. In: What Is MongoDB? | MongoDB [online]. ©2018 [cit. 2018-04-15]. Dostupné z: <https://www.mongodb.com/what-is-mongodb>
- [33] SQL Server Documentation. In: SQL Server Documentation | Microsoft Docs [online]. 2018 [cit. 2018-04-16]. Dostupné z: <https://docs.microsoft.com/en-us/sql/sql-server/sql-server-technical-documentation?view=ssdt-18vs2017>
- [34] Administrator Settings. In: OpenConf Configuration [online]. [cit. 2018-04-20]. Dostupné z: <https://www.openconf.com/documentation/configuration.php>
- [35] Choosing between .NET Core and .NET Framework for server apps. In: Microsoft Docs [online]. [cit. 2018-04-21]. Dostupné z: <https://docs.microsoft.com/en-us/dotnet/standard/choosing-core-framework-server>
- [36] About. In: YAML Ain't Markup Language [online]. [cit. 2018-04-21]. Dostupné z: <http://yaml.org/about.html>
- [37] Conference Management Using EasyChair. In: EasyChair [online]. [cit. 2018-04-23]. Dostupné z: <https://easychair.org/conferences.cgi>
- [38] OpenConf Portfolio. In: OpenConf [online]. Zakon Group, ©2004-2018 [cit. 2018-04-23]. Dostupné z: <https://www.openconf.com/portfolio/>
- [39] OpenConf Demo. In: OpenConf [online]. Zakon Group LLC., ©2004-2018 [cit. 2018-04-23]. Dostupné z: <https://www.openconf.com/demo/>
- [40] Plagiarism Detection Software | iThenticate. In: iThenticate [online]. Zakon Group LLC., 2017 [cit. 2018-04-23]. Dostupné z: <http://www.ithenticate.com/>
- [41] Homepage. PHP: Hypertext Preprocessor [online]. ©2001-2018 [cit. 2018-04-25]. Dostupné z: <http://php.net/>

- [42] Homepage. MySQL [online]. Oracle Corporation, ©2018 [cit. 2018-04-25]. Dostupné z: <https://www.mysql.com/>
- [43] Homepage. MariaDB.org - Supporting continuity and open collaboration [online]. MariaDB Foundation, ©2018 [cit. 2018-04-25]. Dostupné z: <https://mariadb.org/>
- [44] Homepage. TinyMCE | The Most Advanced WYSIWYG HTML Editor [online]. MariaDB Foundation, ©2018 [cit. 2018-04-25]. Dostupné z: <https://www.tinymce.com/>
- [45] Homepage. CKEditor Ecosystem - rich text editors and collaborative editing solutions [online]. CKSource sp. z o.o. sp.k, 2018 [cit. 2018-04-25]. Dostupné z: <https://ckeditor.com/>
- [46] What is Linux? | Linux.com | The source for Linux information. LinuxCom: News for the Open Source Professional [online]. The Linux Foundation, ©2016 [cit. 2018-04-26]. Dostupné z: <https://www.linux.com/what-is-linux>
- [47] The leading operating system for PCs, IoT devices, servers and the cloud | Ubuntu. Ubuntu [online]. Canonical, ©2018 [cit. 2018-04-26]. Dostupné z: <https://www.ubuntu.com/>
- [48] Homepage. Angular CLI [online]. Google, ©2010-2016 [cit. 2018-04-26]. Dostupné z: <https://cli.angular.io/>
- [49] Homepage. JSON Web Tokens - jwt.io [online]. Zakon Group, ©2004-2018 [cit. 2018-04-26]. Dostupné z: <https://jwt.io/>
- [50] Learn About ASP.NET MVC. In: ASP.NET MVC | The ASP.NET Site [online]. Microsoft, ©2018 [cit. 2018-04-27]. Dostupné z: <https://www.asp.net/mvc>
- [51] Dropbox Paper. In: Dropbox Paper – Dropbox [online]. 2015 [cit. 2018-04-29]. Dostupné z: <https://www.dropbox.com/help/paper>

## **A CD obsahující**

1. Zdrojový kód aplikace.
2. Elektronickou verzi této závěrečné práce.